



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO TÉCNICO EN INFORMÁTICA DE GESTIÓN

Título del proyecto:

MOVILIDAD INTERNACIONAL

Alumno: Lorenzo Fidalgo Allo

Tutor: Jesús Villadangos Alonso

Pamplona, 09/02/2014

ÍNDICE

1. INTRODUCCIÓN	2
1.1 NECESIDAD DE OFERTAR TRABAJOS EN LA WEB, WEBS EXISTENTES	2
1.2 NECESIDADES NO CUBIERTAS ACTUALMENTE EN LA WEB	2
1.3 RESUMEN DEL TRABAJO REALIZADO DESTACANDO CARACTERÍSTICAS DE ESPECIAL INTERÉS	3
<i>Mapa interactivo con geolocalización de las empresas.....</i>	<i>4</i>
<i>Enlace directo con Facebook. Zona de comentarios y plugins sociales</i>	<i>4</i>
<i>Bolsas de empresas y trabajadores.....</i>	<i>5</i>
<i>Perfil.....</i>	<i>5</i>
2. HERRAMIENTAS	6
• HTML 5 y CSS3.....	6
• JavaScript.....	7
• AJAX.....	8
• SQL.....	8
• PHP.....	9
• Photoshop.....	9
• Host y Dominio	9
3. REQUISITOS DEL SISTEMA.....	10
3.1 REQUISITOS.....	10
3.1.1 Requisitos funcionales.....	10
3.1.2 Requisitos no funcionales.....	16
3.2 DIAGRAMAS DE CASOS DE USO	16
3.2.1 Actores.....	16
3.2.2 Casos de uso de los usuarios.....	17
4. ARQUITECTURA DEL SISTEMA.....	20
4.1 CAPA DE PRESENTACIÓN:.....	20
4.2 CAPA DE NEGOCIO O LÓGICA DE LA APLICACIÓN:	31
4.3 CAPA DE PERSISTENCIA O DATOS:	37
5. ANÁLISIS Y DISEÑO	45
5.1 MODELO DE DATOS.....	45
5.1.1 Modelo entidad-relación	46
5.1.2 Modelo relacional.....	47
5.2 DIAGRAMA DE CLASES	48
5.3 DISEÑO DE INTERFACES	49
6. IMPLEMENTACIÓN	52
7. PRUEBAS Y RENDIMIENTO DEL SISTEMA	61
8. DESPLIEGUE.....	62
9. METODOLOGÍA DE GESTIÓN DEL PROYECTO	63
10. CONCLUSIONES	64
11. LÍNEAS FUTURAS.....	65
12. BIBLIOGRAFÍA.....	66

1. INTRODUCCIÓN

1.1 Necesidad de ofertar trabajos en la web, webs existentes

Dada la profunda crisis económica global, se ha producido gran destrucción de empleo y, por tanto, las opciones de encontrar un trabajo en la zona geográfica propia se reducen.

El objetivo de este sitio web es proporcionar una herramienta útil a todos aquellos que tengan el deseo de obtener un empleo. Esto es tanto válido para encontrar un empleo en las inmediaciones del área de tu localidad de residencia como para aquellos que consideren movilidad internacional europea, aumentando, así, notablemente sus probabilidades de éxito.

Se trata de lograr un lugar de intercambios de información en internet. Donde aquellos en búsqueda de empleo puedan observar las diferentes posibilidades que ofertan todas las empresas asociadas a la página web, y, a su vez, aquellas empresas que estén registradas en la página web puedan obtener listados de posibles trabajadores que emplear en sus empresas. Así como permitir, por qué no, que todas aquellas personas con trabajo sean capaces de observar el mercado y las diferentes posibilidades de movilidad que dicho sitio web les muestre para que puedan plantearse si les interesa cambiar de trabajo.

Es relativamente sencillo encontrar en la actualidad diferentes páginas web que ofertan trabajos con vacantes que cubrir. Es el caso de sitios como Infojobs, Tecnoempleo o incluso bolsas de trabajo como las del SEPE o el servicio navarro de empleo. Pero el problema reside en que dichos sitios web se limitan a ello. Siguen el patrón “oferto trabajo, abro un buzón al que llegan cientos de CV de diversas personas”. Sin llegar a nada más que eso.

1.2 Necesidades no cubiertas actualmente en la web

Lo que buscamos con movilidad internacional es modificar el paradigma y evitar que sean todos los sitios web de búsqueda de empleo iguales. En el caso que nos ocupa, creamos un lugar fácilmente accesible para todo tipo de usuario en el que se observa de modo muy visual un mapa con todas las empresas que consideran ampliar personal en todo el área de Europa. Evitando limitar a España la oferta y cubriendo, así, un importante nicho del mercado.

Más aún, es nuestro objetivo generar un lugar que proporcione mucha retroalimentación. Comentarios diferentes provenientes de todo aquel que visita el lugar, informando, de este modo, a todos los usuarios de cómo está el mercado, qué zonas son buenas o malas o incluso, por ejemplo, qué empresas sería mejor evitar. Conseguir dicha retroalimentación es una técnica para conseguir implicación del usuario. Evitando que utilice la página web de modo pasivo. Es nuestro interés que se cree implicación social por parte de los usuarios. Comentarios entre los usuarios, hacia el propio sitio web y hablando de las empresas o sus experiencias.

Para realizar esto último no se ha intentado reinventar la rueda, sino aplicar las técnicas que internet pone a nuestra disposición del mejor modo posible. En este caso, he utilizado la red social más potente, Facebook, y he configurado dentro de mi sitio web una zona para que los usuarios puedan conectarse con sus perfiles de la red social y puedan hablar entre ellos, contestarse o incluso enviarse mensajes entre ellos. Abriendo la posibilidad de que sus relaciones continúen en Facebook. Dando ya, a dicha plataforma, la potencia de la socialización, que es en lo que destaca. Preguntas típicas que podrían realizarse entre los empleados podrían ser semejantes a “¿Qué posibilidades hay de trabajar en Polonia dominando el inglés?” o “¿Qué referencias tenéis de la empresa X?”. Por tratar de resumirlo mucho se podría llegar a decir que se trata de una bolsa de empleo muy visual enlazada con la mayor red social del mundo. Un lugar donde reunir diferentes perfiles y que éstos interactúen entre ellos para lograr sus objetivos profesionales.

También debo citar que el modelo de negocio de Movilidad Internacional es diferente a lo que normalmente observamos ya que aquí no se cobraría a las empresas que es lo que hacen empresas como la fuerte infojobs.net. El modelo de negocio de Infojobs y sus semejantes se basa básicamente en dos factores:

Por un lado, los anuncios de diferentes cursos, exámenes y acreditaciones por los que percibe dinero. Básicamente lo llamaríamos publicidad. Por el otro lado, y el que nos afecta en este caso, Infojobs cobra por cada CV que la empresa abre para ver diferentes candidatos. Este último punto, en Movilidad Internacional, resultaría altamente contraproducente, ya que nos interesa el máximo de movimiento posible. Libertad a la hora de mostrar datos y libertad a la hora de elegir. No se buscan cobros por usar el servicio, sino convertir el servicio en un centro concurrido. Se evitan aranceles como el mencionado. Siempre buscaremos crear la posibilidad de que los trabajadores puedan contactar directamente con las empresas a través de email o teléfono. Así como las empresas pueden escoger directamente empleados de la bolsa de trabajadores sin necesidad de tener que esperar a que los empleados acudan a ellas.

Así como el sitio web está orientado actualmente a empleados en búsqueda de trabajo, sería realmente sencillo adaptarlo a otros casos. Por ejemplo, establecer un nexo de unión entre los estudiantes de las diferentes universidades y dichas universidades, pudiendo aplicarlo, por ejemplo, a los programas ERASMUS. Las empresas, en el citado caso, serían las universidades y los trabajadores serían, a su vez, los estudiantes que desean movilidad por Europa o incluso aquellos que ya la han disfrutado y quieren compartir sus opiniones o, simplemente, tener información de primera mano de los programas de movilidad disponibles.

1.3 Resumen del trabajo realizado destacando características de especial interés

Movilidad Internacional es un sitio web preparado para alojar empresas que necesitan mano de obra y trabajadores que quieren la oportunidad de trabajar. Se trata de una página web que quiere servir de enlace a todas aquellas empresas europeas que consideran aumentar personal, con los diferentes trabajadores que buscan o consideran buscar un trabajo. Consiste en la unión de las necesidades de ambos colectivos para generar un tráfico que resuelva los problemas de ambos.

Se busca el registro del máximo número posible de usuarios útiles. A más usuarios, más información, más movilidad.

Algunas de las zonas destacables del trabajo realizado en Movilidad Internacional son las siguientes:

Mapa interactivo con geolocalización de las empresas

En el momento en que una empresa se registra, debe introducir como datos obligatorios que se introducen en la base de datos de la página web, el país, la ciudad y la dirección de la empresa. Tratando estos datos logro enviar una salida de modo que llego a crear un marcador en la localización exacta del mapa. Utilizando para esto el mapa más completo hasta el momento que es el de google maps. Construyo sobre dicho mapa, el cual acoto, inicialmente, a Europa, un marcador por cada empresa registrada. Además, al clicar en un marcador, éste se anima con una animación de rebote y permanece así mientras no se clique en otro marcador o se vuelva a clicar en el mismo, indicando que empresa hemos clicado. En el momento en que clicamos el marcador, se despliega una caja de texto con cierta información. Dentro de dicha caja de texto, encontramos, también, un enlace creado dinámicamente con PHP hacia la página personal de la empresa. Aquella página con unos datos concretos que la empresa decide mostrar, así como una imagen, si es el caso de que la empresa haya decidido mostrar una imagen al resto de usuarios.

Este mapa es completamente interactivo. Se establecen unos valores de acotación y zoom por defecto, pero todo esto es modificable. Se puede navegar perfectamente por el mapa. Es importante señalar, además, que se han creado unas funciones con JavaScript, utilizando el framework JQuery, para agrupar marcadores en función de la cercanía geográfica entre ellos. Esta agrupación ayuda, además, a la búsqueda de las empresas, ya que al clicar en una de las agrupaciones, esto te enviará a un zoom perfecto para ver las empresas del grupo.

Enlace directo con Facebook. Zona de comentarios y plugins sociales

Aprovechando la potencia de la red social más fuerte del momento, he creado en el sitio web una zona donde los usuarios tienen libertad para comentar aquello que les parece de interés. Desde consejos para otros usuarios, hasta críticas de las empresas, pasando por preguntas acerca de empresas o petición de consejos.

Con sólo tener un perfil de Facebook los usuarios pueden enlazar desde la página web a la red social para continuar las conversaciones que les parezcan de interés, ya que tras clicar en un nombre de la zona de comentarios de Movilidad Internacional, eres redirigido al perfil del usuario en sí dentro de Facebook, teniendo la posibilidad de contactar con dicha persona.

Además, he implementado plugins sociales como el “Like” y el “Share”. Esto permite una rápida difusión del sitio web a través de la red social Facebook, ya que cada usuario que clique en éstos, hace que aparezca en el muro personal de su perfil de Facebook una mención a Movilidad Internacional.

Todo lo mencionado en este punto puede realizarse sin tener siquiera que registrarse en Movilidad Internacional.

Bolsas de empresas y trabajadores

Existen dos zonas dentro del sitio web en las cuales se muestran, respectivamente, todos los trabajadores registrados con sus correspondientes datos para ser localizados y todas las empresas registradas con sus respectivos datos para contactarlas. Además, en el caso de las empresas, se ha creado un servicio de mensajería directa con éstas al que se accede con tan sólo un clic. El usuario tiene la posibilidad de enviar un mensaje a la empresa de interés a través de la propia página web.

Perfil

Todo usuario registrado tiene un perfil propio en el que se pueden ver los datos ingresados por uno mismo a la hora de registrarse. En este lugar se pueden actualizar los datos cambiando lo que nos interese. También podemos dar de baja al usuario, si es nuestra voluntad. Claro está, tras introducir la contraseña por motivos obvios de seguridad. El cambio de contraseña por otra también se realiza en esta zona.

En el caso de tratarse el usuario registrado de un gestor de diversas empresas, aquí mismo está habilitada una zona en la cual es posible modificar los datos de las empresas registradas o incluso dar de baja aquellas que nos interese dar de baja. Esto último, por supuesto, es únicamente visible para gestores. Los usuarios registrados como empresas o como trabajadores no tienen acceso a esto. Es más, ni siquiera lo ven en sus perfiles.

2. HERRAMIENTAS

En este proyecto se han utilizado multitud de lenguajes y programas distintos para completar cada uno de los módulos que lo componen ya que se trata de un trabajo que requiere del conocimiento de áreas dispares.

A continuación explico los diferentes lenguajes y programas utilizados desde la parte más externa que sería la maquetación (front-end), hasta la lógica más interna programada con PHP (back-end). Procedo a enumerar los diferentes lenguajes y programas utilizados y el uso que se les ha aplicado.

- **HTML 5 y CSS3**

Se ha utilizado el lenguaje de marcado en su versión más novedosa para conseguir la maquetación de todo el sitio web. Lo cual significa, entre otras cosas, la colocación de todos los elementos del sitio en su correcto lugar, así como elección de la correcta escala de cada uno de ellos y las relaciones que se establecen entre los citados elementos. HTML utiliza el sistema de CSS para establecer estilos a los diferentes elementos creados por HTML, lo cual crea una estrecha relación entre ellos. Por lo general, se podría decir que con HTML se crea un elemento y se le da un identificador o nombre de grupo y con CSS se establece cómo queremos que dichos elementos o grupos sean mostrados a los usuarios. Ejemplos de CSS sería, por ejemplo, el tamaño de letra de los elementos, los colores de dichas letras, los colores de fondo de los elementos creados con HTML y multitud de otras características.

Así como se buscó utilizar HTML 5 que es la versión más actual de HTML, también se ha utilizado CSS3 que es la versión más novedosa de CSS.

HTML significa “HyperText Markup Language” (lenguaje de marcas de hipertexto). CSS significa Cascading Style Sheets (hojas de estilo en cascada).

HTML5 y CSS3: Revolucione el diseño de sus sitios web
El gran libro de HTML5, CSS3 y JavaScript
Mi primer proyecto Web

- **JavaScript**

La tecnología JavaScript se encuentra muy presente en todo el proyecto. JavaScript proporciona facilidad de manejo por parte del usuario y geniales efectos visuales entre otras cosas. También permite crear mapas como el implementado en el proyecto, junto con los propios marcadores de éste, creados también utilizando JavaScript y PHP principalmente. Es más, la función de agrupación de los marcadores, el zoom automático al clicar en estas agrupaciones y las animaciones del mapa son, también, creadas mediante programación en JavaScript. Otro de los usos de JavaScript es permitirnos emitir mensajes de aviso al usuario, como en el caso de que intente acceder a alguna zona restringida para la que no tenga permisos de acceso o señalar que un registro se ha completado con éxito o no y muchas ocasiones semejantes.

Hay ciertas ocasiones, como a la hora de enviar los datos para realizar un registro, donde realizamos una comprobación en el propio cliente de que los datos son correctos utilizando esta misma tecnología. Evitando, de este modo, tener que enviar los datos al servidor y después enviarlos de vuelta, acción que resultaría nada eficiente. Además de todo lo señalado, hay otras ocasiones como, por citar una, la redirección automática tras la identificación de usuario en el sitio web, que utiliza, también, la tecnología en cuestión.

He creado a lo largo del proyecto muchas diferentes funcionalidades gracias al lenguaje JavaScript. Lo he realizado tanto con JavaScript a través de frameworks como JQuery para, por ejemplo, crear transiciones entre imágenes como sin frameworks y codeando todo aquello que me ha sido necesario.

Domine JavaScript
El gran libro de HTML5, CSS3 y JavaScript

- **AJAX**

AJAX (Asynchronous JavaScript And XML) es una tecnología en la que está basada una parte importante del trabajo. A simple vista no puede apreciarse para un usuario que no ha trabajado anteriormente con AJAX, pero tiene ciertas ventajas que quería que tuviera la aplicación y por ello las implementé en el proyecto.

En vez de resignarme a una página web corriente, las cuales están formadas por diferentes páginas con mucho código común con hipervínculos entre las diferentes zonas, tomé la decisión de evitar repetición de código HTML creando una estructura que contendría todas las zonas que se mantienen en las diferentes secciones, como son el logotipo con el título, las imágenes de fondo y la zona de navegación. De este modo consigo que mientras el usuario navega por la aplicación web no tienen que volver a cargarse dichas zonas. Se crea un contenedor dinámico que es el que se va modificando a petición del usuario. Un contenedor central que se va cargando de modo dinámico con las zonas según son visitadas. Con todo esto conseguimos minimizar las cargas.

Además de todo lo señalado, también debo mencionar que una vez que el usuario ha cargado una página en este proyecto, aunque pierda internet podría volver a verla, ya que gracias a AJAX no tiene que volver a pedirla al servidor, sino que la carga directamente, puesto que ya la tiene accesible. Esto es sólo un ejemplo de las ventajas de AJAX. Todo esto maximiza la agilidad de la aplicación.

- **SQL**

Debido a la necesidad de almacenar grandes cantidades de datos en el proyecto web, se utiliza una base de datos relacional que almacena todos los datos de interés como son los trabajadores, empresas y gestores de empresas con todos sus respectivos datos. Del mismo modo, también es registrada en dicha base de datos la información que utilizo para crear los marcadores que identifican la localización de las diferentes empresas, como es la latitud y la longitud.

La base de datos ha sido creada en MySQL con Phpmyadmin, que me permite trabajar perfectamente a través del navegador. Tanto para realizar pruebas a lo largo de la construcción del proyecto en un servidor local como en un servidor de pruebas en internet. Para ir probando conforme iba creando el proyecto, he utilizado la plataforma XAMPP, la cual me ha permitido trabajar y probar mi proyecto de modo rápido y efectivo. Las siglas identifican a Apache, MySQL, PHP y Perl. He utilizado todo esto a excepción de Perl, que no me ha sido necesario.

- **PHP**

La mayor parte de la lógica interna es realizada con el lenguaje de programación PHP. Gracias a PHP he sido capaz de crear todas las páginas del sitio web que necesitan acceso a bases de datos, sesiones de usuarios, comprobación de diversos datos y la muestra de las bolsas de trabajadores y empresas entre otras cosas. PHP ha sido la columna vertebral del proyecto, permitiéndome realizar todo aquello que ha sido necesario.

PHP práctico. Fundamentos.

- **Photoshop**

A lo largo del proyecto ha habido algunos momentos en los cuales he debido realizar tratamiento de imágenes. En prácticamente todos los casos he solventado esta necesidad con el programa Photoshop.

A la hora de crear el título del sitio web y juntarlo en una imagen he utilizado dicho programa, así como cuando he cambiado las propiedades de algunas imágenes para hacerlas menos pesadas para, posteriormente, hacerlas formar parte del sitio web. Haciendo esto consigo que los usuarios descarguen imágenes más ligeras pero con una calidad muy semejante. Logrando mayor velocidad de descarga de la página.

- **Host y Dominio**

La página web ha sido alojada en nixiweb.com. Un servidor gratuito. A su vez, logré encontrar un dominio de extensión .tk para la página. He estado realizando las pruebas en www.movilidadint.tk. Más tarde, por motivos que se explican en este mismo documento, me trasladé a www.movilidadinternacional.tk. Esta combinación de host y dominio gratuitos me ha permitido desarrollar sin coste extra el proyecto.

3. REQUISITOS DEL SISTEMA

Los diagramas de casos de uso nos permiten diferenciar los actores que interactúan con nuestra aplicación, las relaciones entre ellos y las acciones que puede realizar cada uno dentro del sistema. Este tipo de diagramas son fácilmente comprensibles tanto por clientes como por usuarios y representan los requisitos funcionales del sistema.

Los diagramas de casos de uso tienen tres elementos:

Actores: Son los usuarios del sistema. En el caso concreto de nuestro proyecto, los actores son los usuarios de la aplicación. Los actores representan un rol, que puede desempeñar alguien que necesita intercambiar información con el sistema.

Casos de uso: Un caso de uso describe una forma concreta de utilizar parte de la funcionalidad de un sistema. La colección de todos los casos de uso describe toda la funcionalidad del sistema.

Comunicación entre actores y casos de uso: Cada actor ejecuta un número específico de casos de uso en la aplicación. Por eso decimos que hay comunicación entre actores y casos de uso.

3.1 Requisitos

3.1.1 Requisitos funcionales

Procedo a describir los diferentes requisitos funcionales que encontramos en la aplicación web clasificados según el tipo de usuario al que afectan.

Todos los usuarios

Apartado	Descripción
Título	Funcionamiento
Propósito	Acceso a la página de funcionamiento donde se explican los procedimientos de la aplicación web
Entrada	-
Proceso	-
Salida	Se muestra al usuario la página de funcionamiento

Apartado	Descripción
Título	Contacto
Propósito	Acceso a la zona de contacto donde se proporciona un formulario a través del cual se puede enviar un mensaje al creador de la aplicación web
Entrada	Asunto y mensaje
Proceso	Se utilizar el servidor del host para enviar el mensaje del usuario al creador de la aplicación web
Salida	En caso de que se rellene correctamente el formulario de envío, se devuelve un mensaje avisando de esto y redirige al usuario a la página principal de la aplicación. En caso de no haber rellenado alguno de los dos campos, avisa al usuario y se mantiene a la espera de que sean rellenados

Apartado	Descripción
Título	Comentarios red social
Propósito	Establecer una zona en la aplicación donde los usuarios puedan realizar sus comentarios. Enlazando esto con Facebook
Entrada	Datos de FB y mensaje a publicar
Proceso	El usuario se conecta a la zona de los comentarios usando para ello su propio perfil de Facebook y desde ese momento puede comentar, enviar mensajes, likes y compartir la página con otros usuarios de Facebook
Salida	La aplicación cambia acorde a la acción realizada por el usuario

Apartado	Descripción
Título	Interacción con los marcadores de empresas en el mapa
Propósito	Instaurar un mapa de Europa que permita interactuar con los diferentes marcadores, representando cada uno de éstos a una empresa y que dichos marcadores permitan al usuario acceder a la página de información de dicha empresa
Entrada	-
Proceso	Los marcadores son creados a través de diferentes funciones de JavaScript, PHP, HTML y CSS. Accediendo a los datos registrados en la base de datos MySQL
Salida	Todos los marcadores animados de las empresas con los enlaces para mostrar la página personal de cada empresa

Usuario anónimo

Apartado	Descripción
Título	Registro
Propósito	Registrarse en el sistema para formar parte de éste
Entrada	Datos múltiples que varían según el tipo de usuario
Proceso	Registro del usuario en la base de datos
Salida	Confirmación de registro o mensajes advirtiendo los errores cometidos al intentar registrarse. Según proceda.

Apartado	Descripción
Título	Autenticación
Propósito	Ser reconocido por el sistema
Entrada	Nombre de usuario y contraseña
Proceso	Comprobación de la validez de los datos en la base de datos
Salida	Redirección a la página principal en caso de acceso exitoso o mensajes advirtiendo los errores cometidos al acceder. Según proceda.

Apartado	Descripción
Título	Recuperación contraseña
Propósito	Recuperación de la contraseña para los usuarios que se hayan registrado pero hayan olvidado la contraseña
Entrada	Nombre de usuario y tipo
Proceso	Se validan los datos y de ser éstos correctos, se envía la nueva contraseña al email asociado al usuario. Si son incorrectos, se informa de esto al usuario y se le redirige a la página principal, previniendo ataques de brute force En caso de no introducir los datos necesarios, se le avisa al usuario para que los introduzca
Salida	Mensaje avisando del estado de la solicitud

Apartado	Descripción
Título	Intento de acceso al listado de Trabajadores
Propósito	Aviso al usuario de que solamente los usuarios registrados e identificados pueden acceder a esta zona
Entrada	-
Proceso	Se impide al usuario acceder a esta zona y se le redirige a la página principal
Salida	Mensaje advirtiendo que únicamente los usuarios registrados e identificados pueden acceder a esta zona y redirección a la página principal

Apartado	Descripción
Título	Intento de acceso al listado de Empresas
Propósito	Aviso al usuario de que solamente los usuarios registrados e identificados pueden acceder a esta zona
Entrada	-
Proceso	Se impide al usuario acceder a esta zona y se le redirige a la página principal
Salida	Mensaje advirtiendo que únicamente los usuarios registrados e identificados pueden acceder a esta zona y redirección a la página principal

Trabajador, Empresa y Gestor de empresas

Apartado	Descripción
Título	Listado de trabajadores
Propósito	Se muestra una tabla ordenada por país de procedencia donde se muestran todos los trabajadores disponibles
Entrada	-
Proceso	Se obtienen los trabajadores registrados de la base de datos de la aplicación
Salida	Listado de los trabajadores de la base de datos

Apartado	Descripción
Título	Listado de empresas
Propósito	Se muestra una tabla ordenada por país de procedencia donde se muestran todas las empresas registradas en la página
Entrada	-
Proceso	Se obtienen las empresas registradas de la base de datos de la aplicación
Salida	Listado de las empresas de la base de datos

Apartado	Descripción
Título	Perfil
Propósito	Área donde se muestran los datos del usuario y donde se pueden cambiar dichos datos, la contraseña e incluso dar de baja al propio usuario
Entrada	Tipo de dato que desea ser cambiado y nuevo dato. En caso de que desee darse de baja al usuario se debe introducir la contraseña del usuario
Proceso	A través de consultas lanzadas contra la base de datos se modifican los datos que interesen al usuario
Salida	Mensaje mostrando el estado de la solicitud del cliente. Se modifica lo pertinente en la base de datos

Apartado	Descripción
Título	Desconexión
Propósito	Destruir la sesión del usuario actual
Entrada	-
Proceso	Se destruye a través de funciones PHP la sesión del usuario actual
Salida	Destrucción de la sesión del usuario y cambio visible en la barra de navegación donde se observa fácilmente que a través de dicha acción se vuelve a convertir el cliente en un usuario anónimo

Gestor de empresas

Apartado	Descripción
Título	Registro de empresas
Propósito	Registrar rápidamente empresas dependientes del usuario gestor
Entrada	Datos de las empresas
Proceso	Se registran en la base de datos los datos de las empresas dependientes del usuario gestor
Salida	Mensaje informando del estado de la creación de la empresa

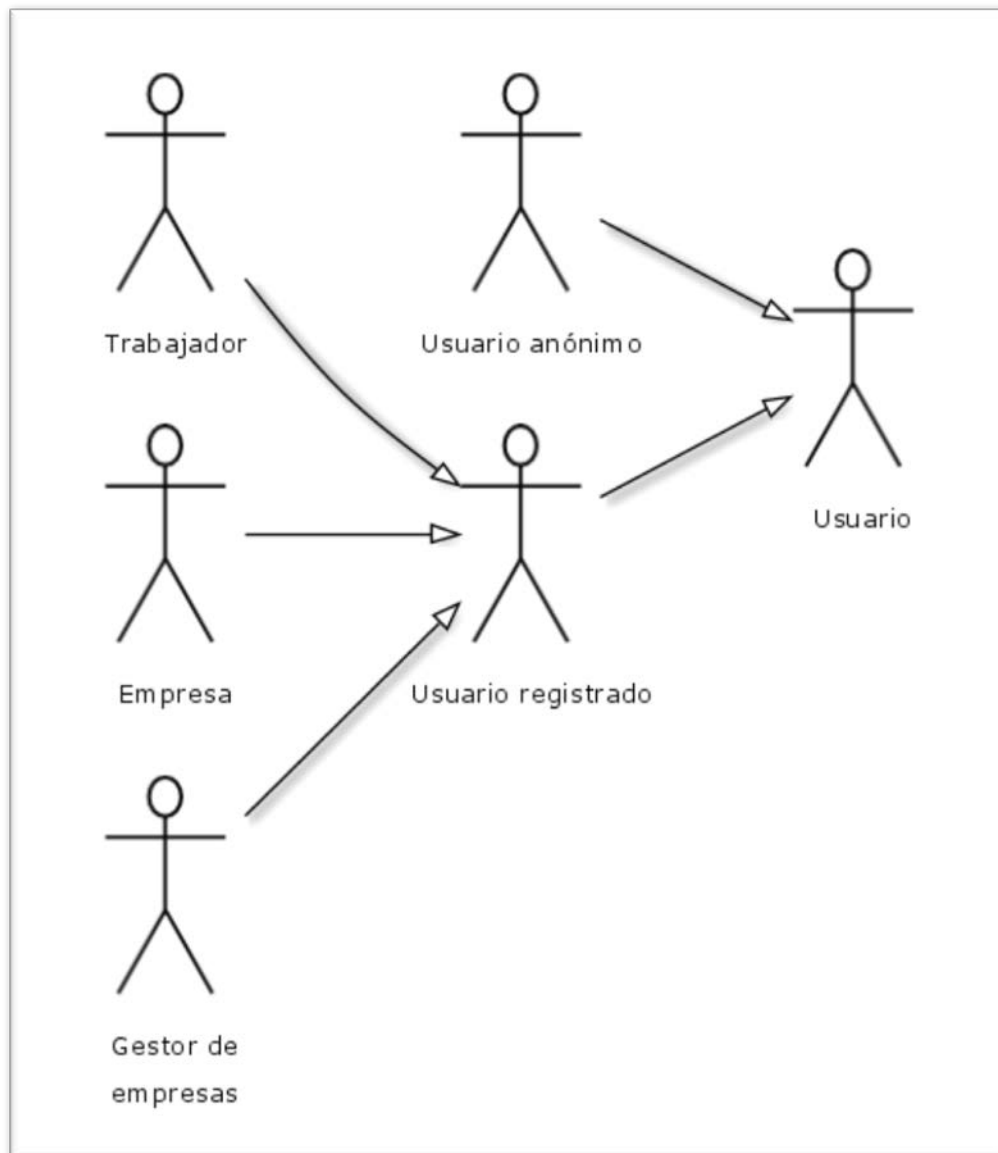
Apartado	Descripción
Título	Perfil de empresas
Propósito	Se permiten modificaciones en las empresas dependientes del gestor, así como la posibilidad de dar de baja éstas.
Entrada	Tipo de dato que desea ser cambiado y nuevo dato. En caso de que desee darse de baja al usuario se debe introducir la contraseña del usuario
Proceso	A través de consultas lanzadas contra la base de datos se modifican los datos que interesen al usuario
Salida	Mensaje mostrando el estado de la solicitud del gestor. Se modifica lo pertinente en la base de datos

3.1.2 Requisitos no funcionales

- ✓ El proyecto debe ser completamente compatible con los navegadores Firefox, Chrome y Safari.
- ✓ Debe optimizarse la velocidad de navegación. Utilizar la tecnología AJAX para mejorar la velocidad de la aplicación.
- ✓ Un usuario nuevo debe ser capaz de utilizar sin error la aplicación web tras, como máximo, 15 minutos de uso.
- ✓ Debe usarse JavaScript para maximizar la interacción con el usuario y mejorar la velocidad de la aplicación.

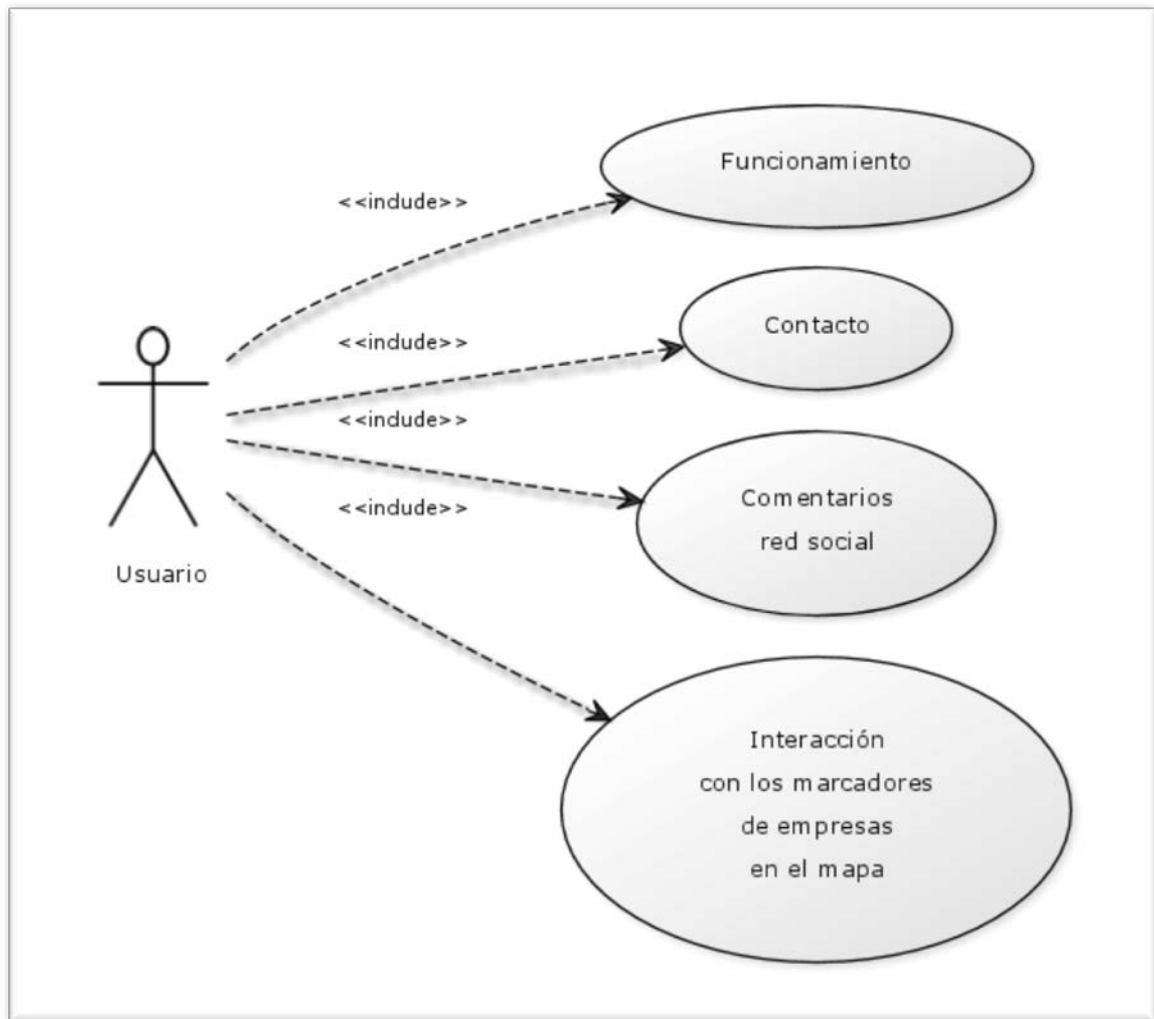
3.2 Diagramas de casos de uso

3.2.1 Actores

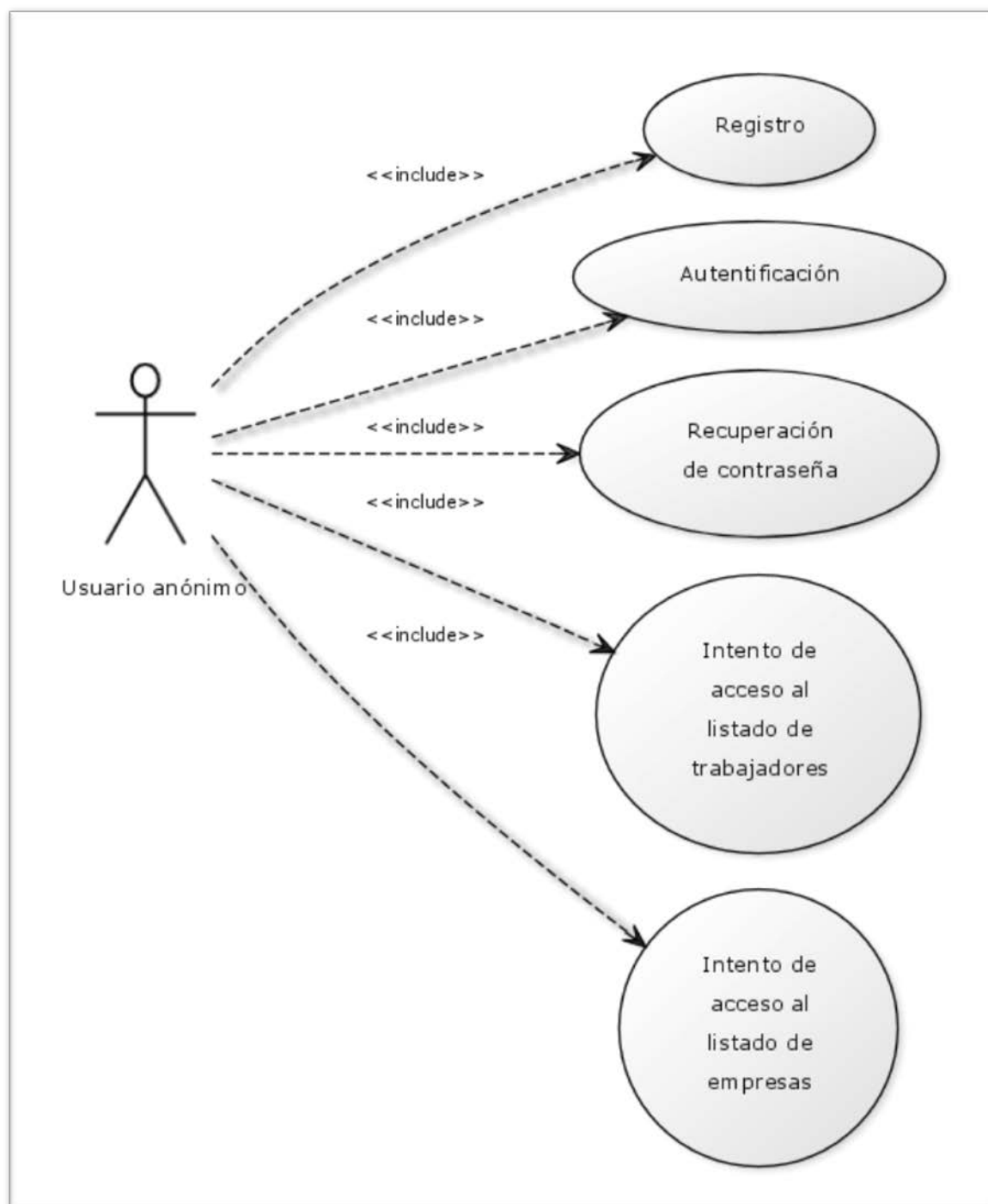


3.2.2 Casos de uso de los usuarios

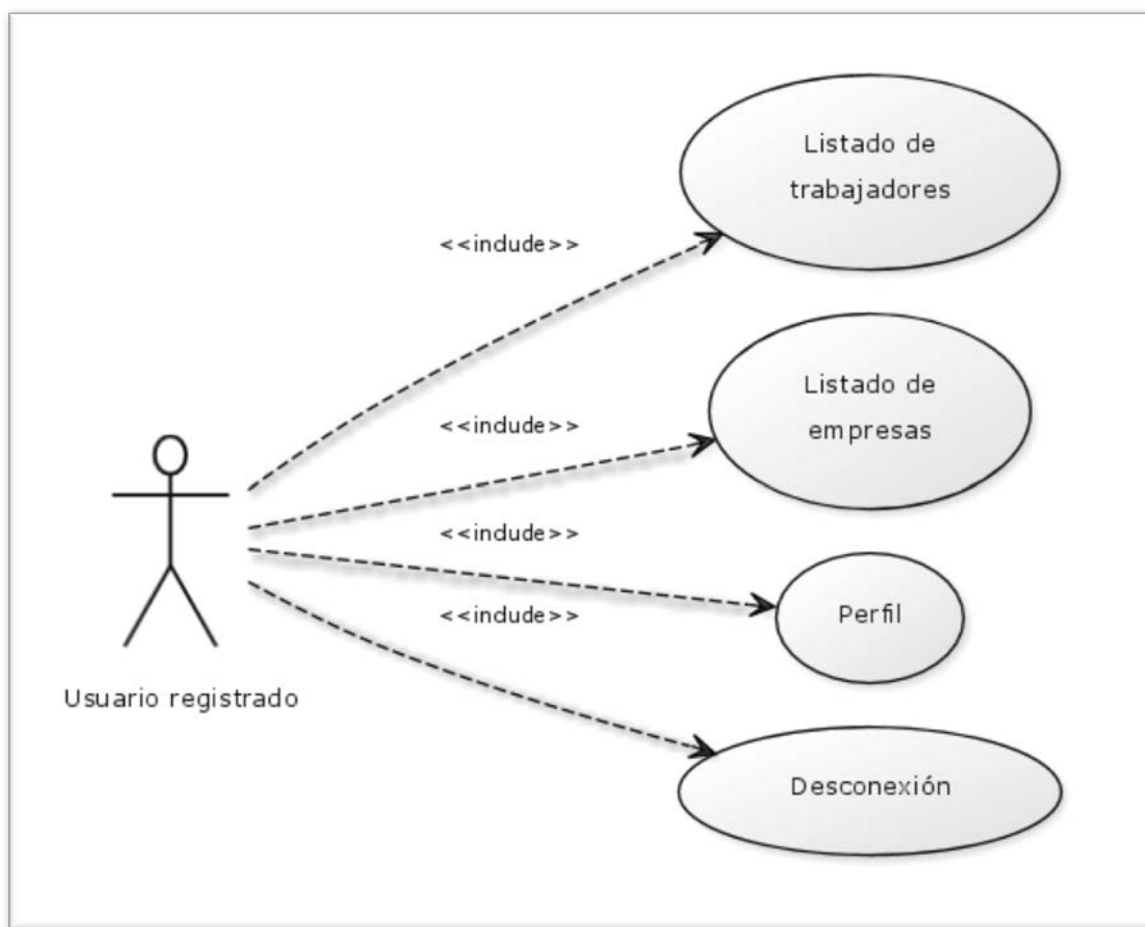
Usuario



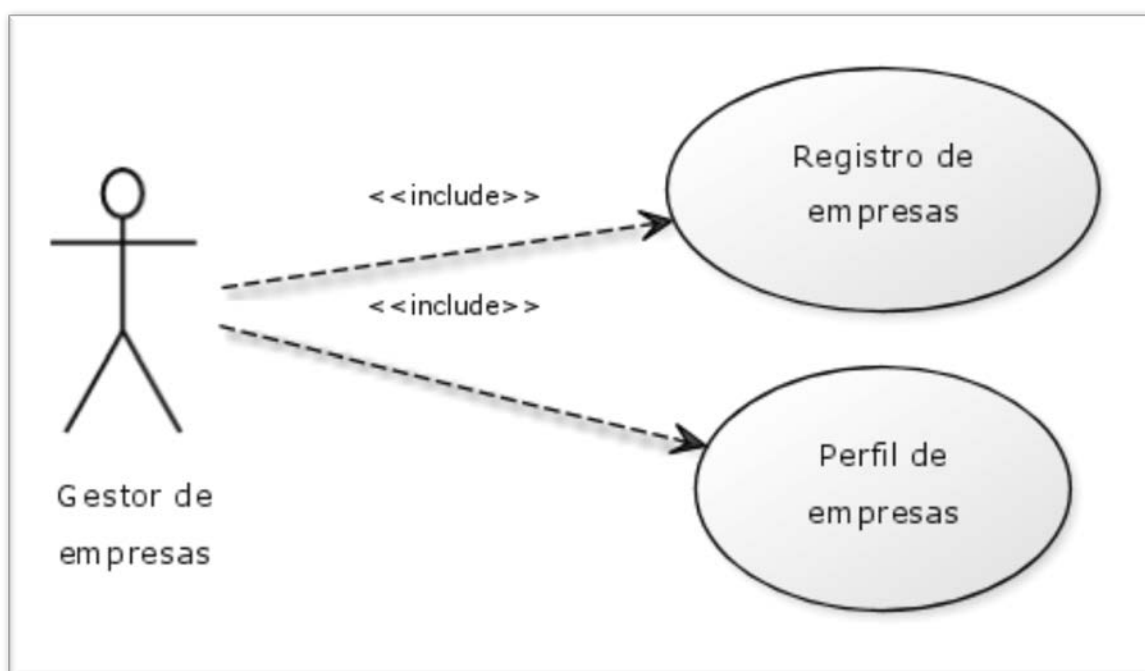
Usuario anónimo



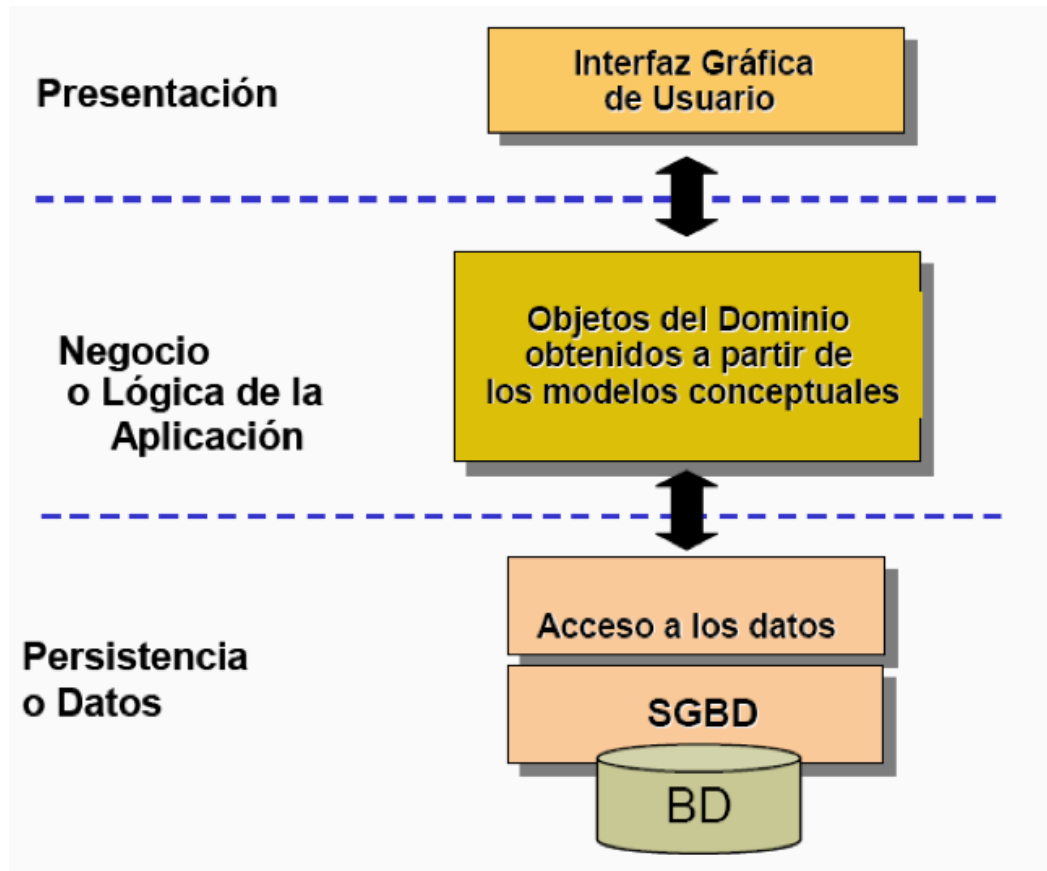
Usuario registrado



Gestor de empresas



4. ARQUITECTURA DEL SISTEMA



4.1 Capa de presentación:

Capa que ve el usuario (también se la denomina "capa de usuario"). Presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. Esta capa se comunica únicamente con la capa de negocio.

En la aplicación que nos ocupa es fácil ver la capa de presentación, ya que toda la interfaz, archivos HTML, CSS e incluso algunos archivos JavaScript que ayudan a la interacción con el usuario pertenecen a ésta. Es todo aquello que utilizamos para comunicarnos con el cliente, tratando de facilitarle la tarea de utilizar nuestra aplicación web.

A continuación muestro algunos ejemplos de codificación de la capa de presentación:

En la siguiente imagen muestro la estructura principal y más externa de la aplicación utilizando HTML. Encontramos el div con id wrapper que es el contenedor de toda la estructura y en los subniveles de éste encontraremos otros contenedores llamados “div”.

```
<div id="wrapper">
  <div id="centrador">
    <div id="marginador">
    </div>

    <header id="cabecera" class="centrado">
      <div id="logo">
        <a href="index.php" id="indexlogo"></a>
      </div>
    </header>

    <nav id="navegacion">
      <ul class="menu">
        <li><a href="funcionamiento.html" id="funcionamiento">Funcionamiento</a></li>
        <li><a href="trabajadores.php" id="trabajadores">Trabajadores</a></li>
        <li><a href="empresas.php" id="empresas">Empresas</a></li>
        <li><a href="contacto.html" id="contacto">Contacto</a></li> <!--Se podría añadir un "Quiénes somos"-->
        <li><a href="perfil.php" id="perfil">Perfil</a></li>
        <li><a href="registroconges.php" id="registroconges">Registro de empresas</a></li>
      </ul>
    </nav>
  </div>
</div>
```

FiguraCódigo1 – Estructura básica HTML

En la siguiente imagen encontramos una comprobación de si el usuario está o no registrado, y dependiendo de esto mostramos u ocultamos cierta información. En este caso concreto, si el usuario está registrado, eliminamos para éste las secciones que le permiten registrarse y logear. Si no está registrado, le indicamos que se identifique.

```
<ul id="logg" class="menu">
<?php
    if ($_SESSION['dentro']=="dentro")
    {
        <li>
            <div id="zonalog">
                <div id="log">
                    <a href="logout.php" id="salir">Desconectarse</a>
                    <!--Hola <strong><?=$_SESSION['usuario']?></strong> | <a href="1
                        <style type="text/css">#rec{display:none;}</style>
                    <script>
                        saliendo();
                    </script>
                </div>
            </div>
        </li>
    }
    <?php
    else
    {
        <li>
            <div id="zonalog">
                <div id="log" class="menu">
                    <a href="acceso.php" id="acceso">Identif&iacute;cate</a>
                </div>
                <div id="rec" class="menu">
```

FiguraCódigo2 – Mostrando HTML diferente dependiendo del usuario

En FiguraCódigo3 se puede observar un div con la clase “clear” que es utilizado para colocar correctamente todos los elementos de la aplicación en su lugar correcto, y tras éste, podemos observar los contenedores donde situaremos la parte central y dinámica de la aplicación que será rellena con AJAX. Además, a partir del div con identificador “like” encontramos el código utilizado para situar en la página web los elementos que enlazan con la red social Facebook. Una vez terminada la colocación, volvemos a utilizar un div de clase “clear” para recolocar los elementos, un “topseparator” para crear una división entre elementos y tras ellos encontramos la sección de footer que es la zona inferior de la aplicación y los cierres del código HTML.

```

<div class="clear">
</div>
<div id="principal" class="centrado">
  <div id="rey">
    <div id="mapa">
    </div>
    <div id="like">
      <div class="fb-like" data-href="http://movilidadint.tk"
        data-layout="standard" data-action="like" data-show-faces="true" data-share="true"></div>
    </div>
    <div class="fb-comments" data-href="http://example.com/movilidadinternacionalf"
      data-numposts="4" data-order-by="reverse_time" data-colorscheme="light"></div>
    <div class="clear">
    </div>
    <!--<div id="comentarios">

    </div-->
    <div class="clear">
    </div>
  </div>
</div>
<div class="clear">
</div>
<div class="topseparator">
</div>
</div>
</div>
<footer>
  Creado por Lorenzo Fidalgo Allo. Copyright&copy;
</footer>
</body>
</html>

```

FiguraCódigo3 – Dibs de colocación, FB y cierre de contenedores

En la siguiente imagen encontramos el código HTML que codifica la interfaz de registro de trabajadores. Encontramos todos los campos necesarios para registrar al usuario trabajador y enviamos la acción de registro a “traregistro.php” donde se realizará la lógica de negocio.

```
<div id="trarecord">
  <div class="cartel">TRABAJADOR
  </div>
  <form id="traform" method="post" action="traregistro.php">

    <label>Nombre de usuario*</label>
    <input name="usuario" type="text">

    <label>DNI*</label>
    <input name="dni" type="text">

    <label>Nombre*</label>
    <input name="nombre" type="text">

    <label>Primer apellido*</label>
    <input name="apellido1" type="text">

    <label>Segundo apellido (si tiene)</label>
    <input name="apellido2" type="text">

    <label>Contraseña de la cuenta*</label>
    <input name="password" type="password" maxlength="30">

    <label>Repite la contraseña*</label>
    <input name="password2" type="password" maxlength="30">

    <label>Email*</label>
    <input name="email" type="text">

    <label>Teléfono</label>
    <input name="telefono" type="text">

    <label>País</label>
    <input name="pais" type="text">

    <label>Ciudad</label>
    <input name="ciudad" type="text">

    <label>Nivel de estudios</label>
    <input name="estudios" type="text">

    <label>Sector</label>
    <input name="sector" type="text">

    <label>Otra información</label>
    <input name="infoextra" type="text">
    <br>

    <input id="trasubmit" name="enviar" type="submit" value="Enviar">
    <br>
    Los elementos marcados con * son obligatorios.

  </form>
```

FiguraCódigo4 – Formulario de registro de trabajadores

Y algunos pantallazos de codificación de estilos CSS:

En la siguiente imagen observamos parte del inicio de la codificación de los CSS. Es una parte minúscula del total de la codificación de estilos (CSS) pero sirve como representación. Aquí vemos la importación desde el servidor de google de diferentes estilos de letra, estableciendo por defecto en todo el documento el tipo 'Bitter'. Además, podemos encontrar otras características, como el establecimiento en todo el documento del padding y los márgenes por defecto a 0 píxeles, para modificarlos e implementarlos según interese a cada zona de la aplicación. Todo esto es solamente algunos de los estilos que podemos ver que están siendo establecidos en esta imagen.

```
@import url(http://fonts.googleapis.com/css?family=Bitter&subset=Latin, Latin-ext);
@import url(http://fonts.googleapis.com/css?family=Droid+Sans);
@font-face
{
    font-family: 'Bitter', serif;
}
*
{
    -moz-box-sizing: border-box;
    -webkit-box-sizing: border-box;
    box-sizing: border-box;

    margin: 0px;
    padding: 0px;

    /*-webkit-backface-visibility:visible;
    -moz-backface-visibility:visible;
    backface-visibility:visible;
    overflow:visible;
    -webkit-overflow:visible;
    -moz-overflow:visible;*/

    font-family: 'Bitter', serif;
}
.clear
{
    clear: both;
}
table
{
    margin:auto;
    text-align:center;
}
.listadotraemp #submit
{
    height: 35px;
    line-height: 35px;
    font-size:1em;
    cursor:pointer;
}

header, section, footer, aside, nav, article, figure, figcaption, hgroup, div
{
    display: block;
}
body
{
    text-align: center;
    overflow-x: hidden;
}
.centrado
{
    margin-left: 2%;
    margin-right: 2%;
}
#wrapper
{
    width: 100%;
    min-width: 960px;
    background-image: url("images/city.jpg");
    background-attachment: fixed;
}
```

FiguraCódigo5 – Zona inicial del establecimiento de los estilos

En la siguiente imagen seguimos observando la codificación de diferentes estilos, podemos ver el establecimiento del estilo del menú de navegación. Concretamente de anchuras, márgenes, paddings, bordes, e incluso el estilo del listado. También se codifican aquí los estilos correspondientes a los elementos del listado. Además de todo esto, también encontramos parte de la codificación de la zona de log.

```
ul.menu
{
    /*float: left;*/
    width: auto;
    list-style: none;
    margin: 0;
    padding: 0;
    border-left: 1px solid #7C7C7C;
}

ul.menu li
{
    display: inline;
    float: left;
}
ul.menu li a:link, ul.menu li a:visited
{
    padding: 0.3em;
    display: block;
    text-decoration: none;
    color: #FFF;
    border-top: 1px solid #7C7C7C;
    border-right: 1px solid #7C7C7C;
    border-bottom: 1px solid #9C9C9C;
}

ul.menu li a:hover
{
    background: rgb(199, 88, 102);
}

ul.menu li a
{
    font-size: 1.2em;
    height: 2.2em;
}

#logg
{
    float: right;
}
#logg.menu li a
{
    color: #99ff99;
}

#logdentro
{
    float: right;
    border: 0;
    max-width: 18%;
    overflow: hidden;
    font-size: 0.7em;
    display: none;
}
#logdentro.menu li
{
    color: #99ff99;
    margin-right: 5px;
}
ul.menu li a#logout
{
    font-size: 0.9em;
    height: auto;
    border: none;
    float: right;
}
```

FiguraCódigo6 – Ejemplo de codificación de estilos con CSS

En FiguraCódigo6 está codificado el estilo correspondiente al mapa del proyecto. Donde establecemos que tendrá una anchura del 100% en relación a su contenedor padre, una altura de 500px, unos márgenes concretos respecto a los elementos adyacentes y, además, dicho mapa proyectará una sombra concreta.

```
#mapa
{
  width: 100%;
  height: 500px;
  /*border: 5px solid;*/
  box-shadow: 4px 7px 4px 0px #222;
  margin-top: 5px;
  margin-bottom: 60px;
}
```


FiguraCódigo7 – Creación del estilo correspondiente al mapa de la web


Gracias a los cuales conseguimos resultados como estas interfaces:











Movilidad Internacional

¡Muévete!

FuncionamientoTrabajadoresEmpresasContactoPerfilRegistro de empresasDesconectarse

Perfil

Nombre de usuario:	Loren
Email:	loren@gmail.com
Organización:	UPNA
Teléfono:	609803086
Información extra:	Front-End software developer
Registrado el:	2013-12-18 13:31:32


Actualizar datos

Organización -

Nuevo dato:

Cambiar dato

Darse de baja



30

upna
Universidade
Pública de Navarra
Nafarroako
Unibertsitatea Publikoa
Todos los derechos reservados
Eskubide guztiak erresalbatu dira

4.2 Capa de negocio o lógica de la aplicación:

Zona en la cual residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

Pertenecen a esta zona todas las funciones PHP o JavaScript con las que programamos nuestra aplicación para conseguir funcionalidades tales como la creación de usuario, listado de empresas, modificación de datos o geolocalización de empresas entre otras.

Codificación de algunas partes de la lógica de negocio:

En la siguiente imagen se plasma parte de la lógica de negocio necesaria para registrar en la base de datos una empresa nueva. Este código está en lenguaje PHP y encontramos zonas importantes como el `“sesión_start();”` método con el que controlamos las sesiones de los usuarios, `“include(‘acceso_db.php’)”` que es la importación del código necesario para conectarnos a la base de datos, que incluye, entre otras cosas, el nombre de la base de datos y la contraseña y nombre del usuario de la base de datos.

Además de todo lo señalado, también vemos en FiguraCódigo8 la comprobación de que el correo es correcto, que no ha introducido campos vacíos e incluso un modo de prevenir código malicioso por parte de un usuario de la aplicación, eliminando caracteres que pudieran provocar problemas en el servidor.


```

<?php
session_start();
include('acceso_db.php'); // incluimos el archivo de conexión a la Base de Datos

/*DADO QUE UTILIZARÉ AJAX, QUITO EL (isset($_POST['enviar'])) */
//if(isset($_POST['enviar'])) // comprobamos que se han enviado los datos desde el formulario
//{
    // creamos una función que nos permita validar el email
    $correo=$_POST['email'];

    function checkEmail($correo)
    {
        $reg = "#^([a-z\d][\.\-\_]*)([a-z0-9])+\@([a-z\d][\.\-\_]*)([a-z\d])+\.[a-z\d]{2,6}$#i";
        return preg_match($reg, $correo);
    }

    if(empty($_POST['nombre']) || empty($_POST['pais']) || empty($_POST['ciudad']) || empty($_POST['direccion']))
    {
        /*echo "Te faltan por introducir campos obligatorios";*/
        ?>
        <script>
            alert("Te faltan por introducir campos obligatorios");
        </script>
        <?php
    }
    elseif( (!checkEmail($correo)) && (strlen(trim($correo)) <> 0 ) )
    {
        // validamos que el email ingresado sea correcto
        /*echo "El email ingresado no es válido."; */
        ?>
        <script>
            alert("El email ingresado no es válido.");
        </script>
        <?php
    }
    else
    {
        // "limpiamos" los campos del formulario de posibles códigos maliciosos

        $cif = mysql_real_escape_string($_POST['cif']);
        $nombre = mysql_real_escape_string($_POST['nombre']);
        $email = mysql_real_escape_string($_POST['email']);
        $telefono = mysql_real_escape_string($_POST['telefono']);
        $pais = mysql_real_escape_string($_POST['pais']);
        $ciudad = mysql_real_escape_string($_POST['ciudad']);
        $direccion = mysql_real_escape_string($_POST['direccion']);
        $sector = mysql_real_escape_string($_POST['sector']);
        $infoextra = mysql_real_escape_string($_POST['infoextra']);

        // comprobamos que el usuario ingresado no haya sido registrado antes

        if((strlen(trim($pais)) < 2) || (!preg_match("/^[a-zA-Z ñáéíóúÁÉÍÓÚ]+$/",$pais)))
        {
            /*echo "Comprueba los datos del campo país"; */
            ?>
            <script>
                alert("Comprueba los datos del campo país");
            </script>
            <?php
        }
        elseif((strlen(trim($ciudad)) < 2) || (!preg_match("/^[a-zA-Z ñáéíóúÁÉÍÓÚ]+$/",$ciudad)))
        {
            /*echo "Comprueba los datos del campo ciudad"; */
            ?>
            <script>
                alert("Comprueba los datos del campo ciudad");
            </script>
            <?php
        }
    }
}

```

FiguraCódigo8 – Código de registro de una empresa

En la imagen siguiente muestro parte del código para modificar los datos que nos introduce el usuario y convertirlos de modo que podamos geolocalizar las empresas en el mapa de la aplicación. Elimino los caracteres extraños y los convierto en aquello que me interese para más tarde pasar dichos datos convertidos a otra función, que será la que geolocalice las empresas.

```
elseif((strlen(trim($direccion)) < 2))
{
    /*echo "Comprueba los datos del campo dirección"; */
    ?>
    <script>
        alert("Comprueba los datos del campo dirección");
    </script>
    <?php
}

else
{
    //Hacemos la llamada al geocoding para obtener la latitud y la longitud
    $codd = str_replace(" ", "+", $direccion);
    $codd = str_replace("a", "", $codd);
    $codc = str_replace(" ", "+", $ciudad);
    $codc = str_replace("a", "", $codc);
    $codp = str_replace(" ", "+", $pais);
    $codp = str_replace("a", "", $codp);
    $jsondec=json_decode(file_get_contents('http://maps.googleapis.com/maps/api/geocode/json?address='.$codd.'+'.$codc.'+'.$codp.'&sensor=false'),true);
    $lat = $jsondec["results"][0]["geometry"]["location"]["lat"];
    $lng = $jsondec["results"][0]["geometry"]["location"]["lng"];

    if((strlen(trim($lat)) < 2) || (strlen(trim($lng)) < 2))
    {
        /*echo "Comprueba los datos de la localización de la empresa"; */
        ?>
        <script>
            alert("Comprueba los datos de la localización de la empresa");
        </script>
        <?php
    }
}
```

FiguraCódigo9 – Comprobaciones para la geolocalización

En la siguiente imagen se muestra la comprobación de que el usuario y contraseña introducidos en el caso de un trabajador sean correctos, y una vez validados los datos, cogemos aquellos datos que nos interesen de la base de datos para trabajar con dicho usuario. Por ejemplo el código que identifica de modo unívoco a dicho trabajador.

```
<?php
session_start();
include('acceso_db.php');

//if(isset($_POST['enviar'])) // comprobamos que se hayan enviado los datos del formulario
//{
    $vusuario=$_POST['usuario'];
    $vpassword=$_POST['password'];
    $vtipo=$_POST['tipo'];
    $_SESSION['tipo']=$_POST['tipo'];
    //if(empty($_POST['usuario']) || empty($_POST['password'])) // comprobamos que los campos usuarios_nombre y usuario_clave no estén vacíos
    if(empty($vusuario) || empty($vpassword))
    {
        >
        <div id="logueo">
            <a href="acceso.php" id="acceso">Identifícate</a>
        </div>
    <?php
    }
    else
    {
        // "limpiamos" los campos del formulario de posibles códigos maliciosos
        $usuario_nombre = mysql_real_escape_string($_POST['usuario']);
        $usuario_clave = mysql_real_escape_string($_POST['password']);
        $usuario_clave = md5($usuario_clave);
        if ($vtipo == "trabajador")
        {
            // comprobamos que los datos ingresados en el formulario coincidan con los de la BD
            $sql = mysql_query("SELECT dni, usuario, pass FROM trabajador WHERE usuario='".$usuario_nombre.'" AND pass='".$usuario_clave.'");

            if($row = mysql_fetch_array($sql))
            {
                $_SESSION['dni'] = $row['dni']; // creamos la sesion "usuario_id" y le asignamos como valor el campo usuario_id
                $_SESSION['usuario'] = $row["usuario"]; // creamos la sesion "usuario_nombre" y le asignamos como valor el campo usuario_nombre
                $_SESSION['dentro']="dentro";
            }
            >
            <a href="logout.php" id="salir">Desconectarse</a>
        }
    }
}
```

FiguraCódigo10 – Comprobaciones de logueo de un trabajador

Análogamente a la imagen anterior, aquí encontramos la validación en la base de datos de un usuario empresa y uno gestor. Generando, a su vez, los datos que nos serán necesarios para trabajar con dichos usuarios. Además, podemos observar que dependiendo de qué tipo de usuario se trate, mostraremos en la página web unos contenidos u otros.

```
elseif ($vtipo == "empresa")
{
    // comprobamos que los datos ingresados en el formulario coincidan con los de la BD
    $sql = mysql_query("SELECT idempresa, usuario, pass FROM empresa WHERE usuario='".$usuario_nombre.'" AND pass='".$usuario_clave.'");

    if($row = mysql_fetch_array($sql))
    {
        $_SESSION['idempresa'] = $row['idempresa']; // creamos la sesion "usuario_id" y le asignamos como valor el campo usuario_id
        $_SESSION['usuario'] = $row["usuario"]; // creamos la sesion "usuario_nombre" y le asignamos como valor el campo usuario_nombre
        $_SESSION['dentro']="dentro";
        $_SESSION['esempresa']="empresa";
        ?>
        <a href="logout.php" id="salir">Desconectarse</a>
        <?php

        //AQUI ES DND TIENE QUE IR EL CÓDIGO CON EL QUE QUITAR "RECORDAR CONTRASEÑA" Y "REGISTRO"
        ?>
        <style type="text/css">#recuperar_contrasena{display:none;}</style>
        <style type="text/css">#rec{display:none;}</style>
        <!-- El código inmediatamente siguiente hace que el usuario sea redirigido a una página. En este caso a index.php -->
        <script>
            alert("Bienvenido <?=$_SESSION['usuario']>");
            location.href="http://localhost/movilidad%20internacional/";
        </script>

        <!-- EN EL SERVIDOR WEB ES LA SIGUIENTE -->
        <!--<script>
            location.href="http://movilidadint.tk/index.php";
        </script-->
        <?php
    }
    else
    {

        <script>
            alert("Los datos introducidos no son correctos. Int\u00E9ntelo de nuevo.")
        </script>
        <div id="logueo">
        <a href="acceso.php" id="acceso">Identifícate</a>
        </div>

    }
}
elseif ($vtipo == "gestor")
{
    // comprobamos que los datos ingresados en el formulario coincidan con los de la BD
    $sql = mysql_query("SELECT idgestor, usuario, pass FROM gestor WHERE usuario='".$usuario_nombre.'" AND pass='".$usuario_clave.'");

    if($row = mysql_fetch_array($sql))
    {
        $_SESSION['idgestor'] = $row['idgestor']; // creamos la sesion "usuario_id" y le asignamos como valor el campo usuario_id
        $_SESSION['usuario'] = $row["usuario"]; // creamos la sesion "usuario_nombre" y le asignamos como valor el campo usuario_nombre
        $_SESSION['dentro']="dentro";
        $_SESSION['esgestor']="gestor";
        ?>
        <a href="logout.php" id="salir">Desconectarse</a>
        <?php

        //AQUI ES DND TIENE QUE IR EL CÓDIGO CON EL QUE QUITAR "RECORDAR CONTRASEÑA" Y "REGISTRO"
        ?>
        <style type="text/css">#recuperar_contrasena{display:none;}</style>
        <style type="text/css">#rec{display:none;}</style>
        <!-- El código inmediatamente siguiente hace que el usuario sea redirigido a una página. En este caso a index.php -->
        <script>
            alert("Bienvenido <?=$_SESSION['usuario']>");
            location.href="http://localhost/movilidad%20internacional/";
        </script>
    }
}
```

FiguraCódigo11 – Comprobaciones de logueo de un trabajador

A continuación, muestro el código necesario para dar de baja voluntariamente a una empresa por parte de dicho usuario. Dado que se trata de un tema delicado, se solicita que introduzca de nuevo la contraseña para verificar la autenticidad del usuario.

```
<?php
session_start();
include('acceso_db.php'); // incluimos el archivo de conexión a la Base de Datos

if(empty($_POST['usuario_clave']))
{
    ?>
    <script>
        alert("Debe introducir la contraseña para eliminar esta empresa.");
    </script>
    <?php
}
else
{
    // "limpiamos" los campos del formulario de posibles códigos maliciosos
    $vdeletear = mysql_real_escape_string($_POST['usuario_clave']);
    $vdeletear = md5($vdeletear);
    // Deleteamos los datos
    $sql = mysql_query("select * from empresa where pass='".$vdeletear.'" and idempresa='".$_SESSION['idempresa']."'");

    if(mysql_num_rows($sql) > 0)
    {
        mysql_query("delete from empresa where pass='".$vdeletear.'" and idempresa='".$_SESSION['idempresa']."'");
        ?>
        <script>
            alert("El usuario ha sido eliminado bajo su petición.");
        </script>
        <?php
        session_destroy();
    }
    else
    {
        ?>
        <script>
            alert("La contraseña introducida no es correcta. Inténtalo de nuevo.");
        </script>
        <?php
    }
}
?>
```

FiguraCódigo12 – Código PHP para dar de baja una empresa

En imágenes como esta puede encontrarse el resultado de dicha codificación. Aquí muestro un listado de las empresas registradas con sus datos en base a unos criterios:



Empresas							
Las empresas interesadas en contratar empleados son las siguientes							
Email	Nombre	País	Ciudad	Dirección	Sector	Teléfono	Información extra
Enviar email	Telefonica Berlin	Alemania	Berlin	Französische Strabe 39	Telefonía	605221225	
Enviar email	Telefonica Madrid	España	Madrid	Paseo de la Castellana, 14	Telefonía	912532554	
Enviar email	Dominic	España	Santiago de Compostela	Rúa da República de el Salvador n 1	Textil	605447885	Buscamos ingenieros para mantenimiento. Enviar email.
Enviar email	Juan y Jose	España	Quintanar de la Orden	Calle Obra, 2	Taller de coches	602554858	Buscamos un mozo de almacén residente en la provincia
Enviar email	Fontanería Juan	España	Málaga	Calle Carretera, 14, Málaga	Fontanería	698558745	
Enviar email	UPNA	España	Pamplona	Calle Tajonar, Pamplona	Universidad	948160002	Se precisan los servicios de 4 empleados para mantenimiento de servidores
Enviar email	Cristalerías Cano	España	Pamplona	Calle Benjamin de Tudela, 8 31012 Pamplona, Navarra	Cristalerías	948554225	Precios imbatibles
Enviar email	SKF	España	Tudela	Carretera de Corella, km 15, 31500 Tudela, Navarra	Industria	948 82 29 00	Buscamos 30 empleados de cadena de montaje
Enviar email	Alfonse	France	Paris	Rue Montmartre, 4	Cleaning	685443337	
Enviar email	Telefonica Paris	Francia	Paris	Rue Montmartre, 24	Telefonía	605221335	IT employers required.
Enviar email	Filipos	Greece	Pella	580 05	Ammo	0033254558	Mechanical Engineer required
Enviar email	Telefonica Londres	Inglaterra	Londres	Waterloo Pl, 3	Telefonía	605221335	
Enviar email	Torinos	Italy	Torino	Corso Galileo Ferraris, 3	Textile	669855234	4 dependents required.
Enviar email	Telefonica						Wife married 8 years



4.3 Capa de persistencia o datos:

Aquí residen los datos, y es la encargada de acceder a los mismos. Está formada por la base de datos que realiza todo el almacenamiento de datos. Se reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Los datos, tales como los usuarios registrados, son almacenados en la base de datos. Además, se realizan consultas provenientes desde código PHP de la capa de negocio hasta esta capa para obtener distintos datos, organizarlos, listarlos o incluso modificarlos.

Muestro, a continuación, algunos de los fragmentos de código que nos permiten mantener la persistencia de datos en el proyecto.

Parte de las comprobaciones que realizo en código PHP para validar que los datos introducidos son correctos.

```
<?php
include('acceso_db.php'); // incluimos el archivo de conexión a la Base de Datos

/*DADO QUE UTILIZARÉ AJAX, QUITO EL (isset($_POST['enviar'])) */
//if(isset($_POST['enviar'])) // comprobamos que se han enviado los datos desde el formulario
//{
?>

<?php
$dni = mysql_real_escape_string($_POST['dni']);

// creamos una función que nos permita validar el email
$correo=$_POST['email'];

function checkEmail($correo)
{
    $reg = "#^([a-z\d][\.\-\_])*\.[a-z\d]{0,62}([a-z\d]{2,6})$#i";
    return preg_match($reg, $correo);
}

function es_DNI_NIE_valido($cadena)
{
    //Comprobamos longitud
    if (strlen($cadena) != 9) return false;

    //Posibles valores para la letra final
    $valoresLetra = array(
        0 => 'T', 1 => 'R', 2 => 'W', 3 => 'A', 4 => 'G', 5 => 'M',
        6 => 'Y', 7 => 'F', 8 => 'P', 9 => 'D', 10 => 'X', 11 => 'B',
        12 => 'N', 13 => 'J', 14 => 'Z', 15 => 'S', 16 => 'Q', 17 => 'V',
        18 => 'L', 19 => 'H', 20 => 'C', 21 => 'K', 22 => 'E'
    );

    //Comprobar si es un DNI
    if (preg_match('/^[0-9]{8}[A-Z]$/i', $cadena))
    {
        //Comprobar letra
        if (strtoupper($cadena[strlen($cadena) - 1]) !=
            $valoresLetra[((int) substr($cadena, 0, strlen($cadena) - 1)) % 23])
            return false;

        //Todo fue bien
        return true;
    }

    //Comprobar si es un NIE
    else if (preg_match('/^[XYZ][0-9]{7}[A-Z]$/i', $cadena))
    {
        //Comprobar letra
        if (strtoupper($cadena[strlen($cadena) - 1]) !=
            $valoresLetra[((int) substr($cadena, 1, strlen($cadena) - 2)) % 23])
            return false;

        //Todo fue bien
        return true;
    }

    //Cadena no válida
    return false;
}

// Procedemos a comprobar que los campos del formulario no estén vacíos
$sin_espacios = count_chars($_POST['usuario'], 1);
if(!empty($sin_espacios[32]))
{ // comprobamos que el campo usuario_nombre no tenga espacios en blanco
    /* echo "El campo <em>usuario_nombre</em> no debe contener espacios en blanco.";*/
    ?>
    <script>
        alert("El campo <em>usuario_nombre</em> no debe contener espacios en blanco.");
    </script>
    <?php

```

FiguraCódigo13 – Comprobaciones en PHP para el registro en la base de datos

Continuamos con las comprobaciones en PHP para asegurarnos de que registramos correctamente los datos del usuario en la base de datos.

```
// Procedemos a comprobar que los campos del formulario no estén vacíos
$sin_espacios = count_chars($_POST['usuario'], 1);
if(!empty($sin_espacios[32]))
{ // comprobamos que el campo usuario_nombre no tenga espacios en blanco
    /* echo "El campo <em>usuario_nombre</em> no debe contener espacios en blanco."/; */
    ?>
    <script>
        alert("El campo <em>usuario_nombre</em> no debe contener espacios en blanco.");
    </script>
    <?php
}
elseif(empty($_POST['usuario']))
{ // comprobamos que el campo usuario_nombre no esté vacío
    /* echo "No has ingresado tu usuario. "; */
    ?>
    <script>
        alert("No has ingresado tu usuario.");
    </script>
    <?php
}
elseif(empty($_POST['password']))
{ // comprobamos que el campo usuario_clave no esté vacío
    /* echo "No has ingresado contraseña."; */
    ?>
    <script>
        alert("No has ingresado contraseña.");
    </script>
    <?php
}
elseif($_POST['password'] != $_POST['password2']) { // comprobamos que las contraseñas ingresadas coincidan
    /* echo "Las contraseñas ingresadas no coinciden."; */
    ?>
    <script>
        alert("Las contraseñas ingresadas no coinciden.");
    </script>
    <?php
}
elseif(!checkEmail($correo)) { // validamos que el email ingresado sea correcto
    /* echo "El email ingresado no es válido."; */
    ?>
    <script>
        alert("El email ingresado no es válido.");
    </script>
    <?php
}
elseif (!es_DNI_NIE_valido ($dni))
{
    /* echo "Introduzca correctamente el DNI o NIE. Formato del DNI: 00000000A"; */
    ?>
    <script>
        alert("Introduzca correctamente el DNI o NIE. Formato del DNI: 00000000A");
    </script>
    <?php
}
else
{
    if(empty($_POST['usuario']) || empty($_POST['dni']) || empty($_POST['nombre']) || empty($_POST['apellido1']) || empty($_POST['email']))
    {
        /* echo "Te faltan por introducir campos obligatorios."; */
        ?>
        <script>
            alert("Te faltan por introducir campos obligatorios");
        </script>
        <?php
    }
    else
    {
        // "limpiamos" los campos del formulario de posibles códigos maliciosos
        $usuario = mysql_real_escape_string($_POST['usuario']);
    }
}
```

FiguraCódigo14 – Más comprobaciones en PHP para el ingreso en la BDD

En el siguiente fragmento de código volvemos a evitar código malicioso y evitamos, además, introducir un usuario si ya ha sido registrado su DNI o su nombre de usuario. En dichos casos se le advierte al usuario del error.

```
else
{
    // "limpiamos" los campos del formulario de posibles códigos maliciosos
    $usuario = mysql_real_escape_string($_POST['usuario']);
    //$dni = mysql_real_escape_string($_POST['dni']); //Lo he subido arriba para trabajar antes con él
    $nombre = mysql_real_escape_string($_POST['nombre']);
    $apellido1 = mysql_real_escape_string($_POST['apellido1']);
    $apellido2 = mysql_real_escape_string($_POST['apellido2']);
    $password = mysql_real_escape_string($_POST['password']);
    $password2 = mysql_real_escape_string($_POST['password2']);
    $email = mysql_real_escape_string($_POST['email']);
    $telefono = mysql_real_escape_string($_POST['telefono']);
    $pais = mysql_real_escape_string($_POST['pais']);
    $ciudad = mysql_real_escape_string($_POST['ciudad']);
    $estudios = mysql_real_escape_string($_POST['estudios']);
    $sector = mysql_real_escape_string($_POST['sector']);
    $infoextra = mysql_real_escape_string($_POST['infoextra']);

    // comprobamos que el usuario ingresado no haya sido registrado antes
    $sql = mysql_query("SELECT usuario FROM trabajador WHERE usuario='". $usuario."'");
    $sql2 = mysql_query("SELECT dni FROM trabajador WHERE dni='". $dni."'");

    if(mysql_num_rows($sql) > 0)
    {
        /*echo "El nombre de usuario elegido ya ha sido registrado anteriormente."; */
        ?>
        <script>
            alert("El nombre de usuario elegido ya ha sido registrado anteriormente.");
        </script>
        <?php
    }
    elseif (mysql_num_rows($sql2) > 0)
    {
        /*echo "El dni elegido ya ha sido registrado anteriormente."; */
        ?>
        <script>
            alert("El dni elegido ya ha sido registrado anteriormente.");
        </script>
        <?php
    }
}
```

FiguraCódigo15– Eliminamos código malicioso y evitamos repetir datos

Aquí observamos la encriptación de la contraseña y la posterior introducción del trabajador una vez validados sus datos en la base de datos. Una vez correctamente insertado, se le avisa al usuario de ello y se le redirige a la página principal.

```
else
{
    $usuario_clave = md5($_POST['password']); // encriptamos la contraseña ingresada con md5

    // ingresamos los datos a la BD
    $reg = mysql_query("INSERT INTO trabajador (dni, nombre, apellido1, apellido2, usuario, pass, email,
    telefono, pais, ciudad, nivelestudios, sector, infoextratra, daterec) VALUES ('".$_dni."',
    '".$_nombre."', '".$_apellido1."', '".$_apellido2."', '".$_usuario."', '".$_usuario_clave."',
    '".$_email."', '".$_telefono."', '".$_pais."', '".$_ciudad."', '".$_estudios."', '".$_sector."',
    '".$_infoextra."', NOW() + INTERVAL 6 HOUR)");

    if($reg)
    {
        /*echo "Trabajador registrado correctamente a las ".date("H:i:s",time()+21600); */
        ?>
        <script>
            alert("Trabajador registrado correctamente. Será redirigido a la página principal.");
            location.href="http://localhost/movilidad%20internacional/";
        </script>

        <!-- EN EL SERVIDOR WEB ES LA SIGUIENTE -->
        <!--<script>
            location.href="http://movilidadint.tk/index.php";
        </script>
        -->
        </script>
        <?php
    }
    else
    {
        /*echo "Los datos introducidos no son correctos. Compruébalos."; */
        ?>
        <script>
            alert("Los datos introducidos no son correctos. Compruébalos.");
        </script>
        <?php
    }
}
```

FiguraCódigo16 – Inserción del trabajador en la base de datos

A continuación tenemos el código correspondiente a las comprobaciones y actualización de los datos de un trabajador bajo petición de dicho trabajador.

```
<?php
session_start();
include('acceso_db.php'); // incluimos el archivo de conexión a la Base de Datos

if(empty($_POST['updatear']))
{
    ?>
    <script>
        alert("No has introducido datos");
    </script>
    <?php
}
else
{
    // "limpiamos" los campos del formulario de posibles códigos maliciosos
    $vupdatear = mysql_real_escape_string($_POST['updatear']);

    //HACER QUE SI ES IGUAL LO QUE HAY Y LO QUE SE QUIERE INTRODUCIR NO SE META
    $sql = mysql_query("SELECT ".$_POST['datoup']." FROM trabajador
    WHERE ".$_POST['datoup']." = '".$vupdatear."' and dni='".$$_SESSION['dni']."' ");

    if(mysql_num_rows($sql) > 0)
    {
        ?>
        <script>
            alert("Si quiere modificar los
            | datos, deben ser distintos a los existentes.");
        </script>
        <?php
    }
    else
    {
        // ingresamos los datos a la BD
        $reg = mysql_query("update trabajador set
        ".$_POST['datoup']."='".$vupdatear."' where dni
        = '".$$_SESSION['dni']."'");
        if($reg)
        {
            //echo "Ha registrado correctamente el campo ".$_POST['datoup']."' a las ".
            ?>
            <script>
                alert("Ha registrado correctamente
                el campo elegido.\nSi desea ver los
                cambios, vuelva a entrar en 'Perfil'");
            </script>
            <?php
        }
        else
        {
            ?>
            <script>
                alert("Los datos introducidos
                no son correctos. Compruébalos.");
            </script>
            <?php
        }
    }
}
}
```

FiguraCódigo17 – Actualización de los datos de un trabajador

Aquí muestro la persistencia de datos en estado puro: Una vez registrados los datos de los diferentes usuarios del proyecto y almacenada su información.

idempresa	idusuario	nombre	usuario	pass	email	sector	pais	ciudad	direccion	telefono	infoextraemp	daterec	lat	lng	urlfoto	idgestor
15	15	Dominic	Dominic	c68bda9542111778aa2eeb45c50	dominic@gmail.com	Textil	España	Santiago de Compostela	Rúa da República de el Salvador nº 1	605447885	Buscamos ingenieros para mantenimiento Enstar em...	2013-12-18 12:49:19	42.8745234	-8.5448847	http://www.construccionyreformasmadrid.com/s/cv_m...	NGEL
14	14	Juan y Jose	JJ	865d6e951b40475c3b7625dc143bc0	qdo@hotmail.com	Taller de coches	España	Quintanar de la Orden	Calle Obra 2	602554858	Buscamos un mazo de almacen residente en la provin	2013-12-18 12:46:14	39.5919132	-3.041338	http://us.123f.com/400wm/400/400/yulufu/yulufu1...	NGEL
13	13	Fonteria Juan	fontaj	4adbaae3ab6da00f13be73fa105f	fontaj@gmail.com	Fontaneria	España	Málaga	Calle Carreteria 14, Málaga	698558745		2013-12-18 12:43:31	36.7218779	-4.4247574	http://us.123f.com/400wm/400/400/yulufu/yulufu1...	NGEL
12	12	UPNA	UPNA	005d1821442a1b6d4709daa3c1a21e4c	unavara@unavara.es	Universidad	España	Pamplona	Calle Tignor Pamplona	948160002	Se precisan los servicios de 4 empleados para mant.	2013-12-18 12:40:39	42.8020514	-1.6342085	http://www.navaracnfidencial.com/wp-content/uplo...	NGEL
11	11	Oristalerias Cano	Cano	027719353827711ced756dac3942a6f	ccano@gmail.com	Oristalerias	España	Pamplona	Calle Benjamin de Tudela 8 31012 Pamplona, Navar	948554225	Piecos imbatibles	2013-12-18 12:36:10	42.8064201	-1.6725225	http://us.123f.com/400wm/400/400/madito/madit...	NGEL
10	10	SKF	SKF	1d06d6567339393c126409c216058	skfudota@gmail.com	Industria	España	Tudela	Carretera de Tudela km 3.5. 31500 Tudela, Navar	948.82.23.00	Buscamos 30 empleados de cadena de montaje	2013-12-18 12:33:14	42.9038304	-1.6155001	http://cmi-peru.com/img/proyectos/D-29-mai_2_p...	NGEL
16	16	Erikson Lawyers	Erikson	5101ca5094cab4ee1b7a7eeb5269e72	elw@gmail.com	Abogacia	Noruega	Oslo	Olsens gate 3	605887558	We need 3 lawyers. English required	2013-12-18 13:14:51	59.9179915	10.744353	http://cmi-peru.com/img/proyectos/D-29-mai_2_p...	NGEL
17	17	Alfonse	Alfonse	8d961cf05042596a7b02610c1019d6d	alfonsec@gmail.com	Cleaning	France	Paris	Rue Montmartre 4	685445337		2013-12-18 13:17:30	48.8633017	2.3409022	http://cmi-peru.com/img/proyectos/D-29-mai_2_p...	NGEL
18	18	Geraht	Geraht	b88b6c3233a4c3675155bdc4503a3	geraht@yahoo.com	Software develop	Polonia	Plock	Zygmunta Wolskiego 10	00372855228	IT employers required	2013-12-18 13:21:38	52.5569955	19.8855298	http://it4-top.blogspot.com/-/27p365ZUWw-UR07CzEkQWw...	NGEL
19	19	Filipos	Filipos	6a2c6ebc423700679c32b7a0085413eef	filipo2@gmail.com	Armo	Greece	Pella	580 05	0033254558	Mechanical Engineer required	2013-12-18 13:26:47	40.7632436	22.5260527	http://it4-top.blogspot.com/-/27p365ZUWw-UR07CzEkQWw...	NGEL
20	20	Torinos	Torinos	6854a4ae93351687afc4e2a458ba69	torinos@hotmail.com	Textile	Italy	Torino	Corso Galileo Ferraris, 3	669855224	4 dependents required	2013-12-18 13:29:15	45.0710856	7.6749022	http://it4-top.blogspot.com/-/27p365ZUWw-UR07CzEkQWw...	NGEL
21	21	Telefonica Madrid			telefonicamadrid@telefonica.com	Telefonía	España	Madrid	Paseo de la Castellana, 34	912552554		2013-12-18 13:33:39	40.4280924	-3.6892022	http://www.purtonversiones.com/imagenesversiones/...	7
22	22	Telefonica Paris			telefonicaparis@telefonica.com	Telefonía	France	Paris	Rue Montmartre, 24	605221335	IT employers required	2013-12-18 13:36:32	48.8641447	2.3455618	http://www.purtonversiones.com/imagenesversiones/...	7
24	24	Telefonica Londres			telefonicalondres@telefonica.com	Telefonía	Inglaterra	Londres	Waterloo Pl. 3	605221335		2013-12-18 13:37:49	51.507365	-0.1326505	http://www.purtonversiones.com/imagenesversiones/...	7
25	25	Telefonica Berlin			telefonicaberlin@telefonica.com	Telefonía	Alemania	Berlin	Französische Strabe 33	605221225		2013-12-18 13:38:19	52.515	13.39557	http://www.purtonversiones.com/imagenesversiones/...	7

13:40:19 - 2025-11-19 14:00:00 - 13:40:19

Examinar

Estructura

SQL

Buscar

Insertar

Exportar

Importar

Operaciones

Seguimiento

Mostrando registros 0 - 6 (7 total. La consulta tardó 0.0005 seg)

SELECT * FROM trabajador LIMIT 0, 30

Perfilando

En línea

Editar

Explicar SQL

Crear código PHP

Actualizar

Mostrar : Fila de inicio 0 Número de filas: 30 Cabeceras cada 100 filas

Ordenar según la clave: Ninguna

+ Opciones

T

Editar

Copiar

Borrar

51988543C

Ambrosio

López

Justin

Ambrosio

49546abac70050b4b2644a38964a1d

ambrosioj@hotmail.com

948554525

España

Pamplona

Universitarios

Ingeniería Mecánica

Movilidad nacional total

2013-12-18 12:29:28

T

Editar

Copiar

Borrar

63402069F

Juan Carlos

Fernández

Domingo

JuanCar

760939ac465d9608369330978c0a5b

jcarnc@gmail.com

948545887

España

Cadiz

Licenciatura

Derecho

Movilidad total

2013-12-18 12:27:52

T

Editar

Copiar

Borrar

63402069F

Mate

García

Matte

Matte

a1479c95aa86788951a26c4f91b1c8d4

mattejp@gmail.com

603225987

España

Soria

Universitarios

Medicina

3 años de experiencia

2013-12-18 12:31:05

T

Editar

Copiar

Borrar

63201020C

Frederic

Garc

Frederic

ada099a597f0878704490b1533a06db

frederic@gmail.com

669855447

France

Paris

FP2

Mantenimiento de máquinas

Movilidad total

Francés nativo

2013-12-18 13:42:21

T

Editar

Copiar

Borrar

65446687V

Marc

Butler

Marc

35a698af0367b656dfb21b7a736af

marc@gmail.com

600221447

España

Tanuel

FP2

Psicología

Preferencia trabajo dentro de Aragón

2013-12-18 13:43:52

T

Editar

Copiar

Borrar

65671009V

Alexia

Came

Alexia

42dbac47a4a74633195ad10a00780d9

alexiacame@gmail.com

608955488

España

Huesca

Universitarios

Enseñanza

Magisterio infantil

2013-12-18 13:50:19

T

Editar

Copiar

Borrar

0976717BK

ja

ja

ja

e549c2056c6bd44540ebb476d902702

jesusa@unavara.es

2013-12-18 19:29:49

Marcar todos / Desmarcar todos Para los elementos que están marcados:

Cambiar

Borrar

Exportar

Mostrar : Fila de inicio 0 Número de filas: 30 Cabeceras cada 100 filas

Operaciones sobre los resultados de la consulta

Vista de impresión

Previsualización para imprimir (documento completo)

Exportar

Mostrar gráfico

Crear vista

Guardar esta consulta en favoritos

Etiqueta:

Permitir que todo usuario pueda acceder a este favorito

Guardar esta consulta en favoritos

MySQL - 4.0.13.760 - mysql - phpMyAdmin

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Seguimiento

Mostrando registros 0 - 2 (3 total. La consulta tardó 0.0005 seg)

```
SELECT *
FROM gestor
LIMIT 0, 30
```

Perfando [En línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar: Fila de inicio: 0 Número de filas: 30 Cabeceras cada: 100 Mas

Ordenar según la clave: Ninguna

Opciones

	idgestor	usuario	pass	email	organizacion	telefono	infoextrages	datefec
Editar Copiar Borrar	6	ja	dbe9042cfa2655c30e39e039864	jausun@unacoma.es	a		a	2013-12-18 19:30:47
Editar Copiar Borrar	7	Loren	5a0d4d4baf0b5a671d3d359a9f1f	loren@gmail.com	UPIA	609603086	Front-End software developer	2013-12-18 13:31:32
Editar Copiar Borrar	6	Lorenzo	5ea121ab333fa7b95e44da538bc1	lorenzofai@gmail.com	UPIA	609603086	Front-End software developer	2013-12-18 13:30:25

Marcar todos / Desmarcar todos Para los elementos que están marcados: Cambiar Borrar Exportar

Mostrar: Fila de inicio: 0 Número de filas: 30 Cabeceras cada: 100 Mas

Operaciones sobre los resultados de la consulta

Vista de impresión Previsualización para imprimir (documento completo) Exportar Mostrar gráfico Crear vista

Guardar esta consulta en favoritos

Etiqueta: Permitir que todo usuario pueda acceder a este favorito

Guardar esta consulta en favoritos

5. ANÁLISIS Y DISEÑO

5.1 Modelo de datos

Los modelos de datos son esenciales para el desarrollo de sistemas de información, ya que a través de ellos puede conseguirse la compatibilidad necesaria para manejar grandes cantidades de datos. Además, si se utilizan las mismas estructuras para el almacenamiento y el acceso, distintas aplicaciones podrán compartir información. Un claro ejemplo es un formulario de registro en una página web, donde para nuestro nombre podemos ingresar únicamente letras, pero para nuestra fecha de nacimiento, números con un orden específico (como por ejemplo día, mes y año). Estas restricciones responden a necesidades de la base de datos y deberían ser iguales en dos bases entre las que cabe la posibilidad de compartir información. De otro modo, esto puede llegar a causar múltiples (y, a veces, nada evidentes) problemas.

Como contrapartida de la gran eficiencia que ofrecen, siempre mirando desde la perspectiva de una empresa, se encuentra el alto costo del desarrollo y el mantenimiento de los sistemas y de las interfaces. Si la calidad de los modelos de datos implementados no es buena, se convierten en obstáculos para el crecimiento de una empresa. Generalmente, esto salta a la vista cuando se desea realizar una ampliación de los sistemas pero, en cierto momento, no se invirtió en un buen modelo de datos, utilizando una base de datos ineficiente y mal concebida por lo que en el momento de modernizarla creará muchísimos problemas y acabará resultando más sencillo y barato destruir todo y crearlo desde el principio.

Un modelo de datos, puede interpretarse como un esquema conceptual, lógico y físico:

Conceptual:

Especifica las expresiones permitidas por el modelo mismo, comunica las reglas y definiciones esenciales de los datos a los usuarios.

Muy general y abstracto, visión general del negocio/institución.

Lógico:

Describe la semántica de tablas y columnas, clases orientadas a objetos, etc. Está representada por una tecnología de manipulación en particular (como SQL).

Versión completa que incluye todos los detalles acerca de los datos.

Físico:

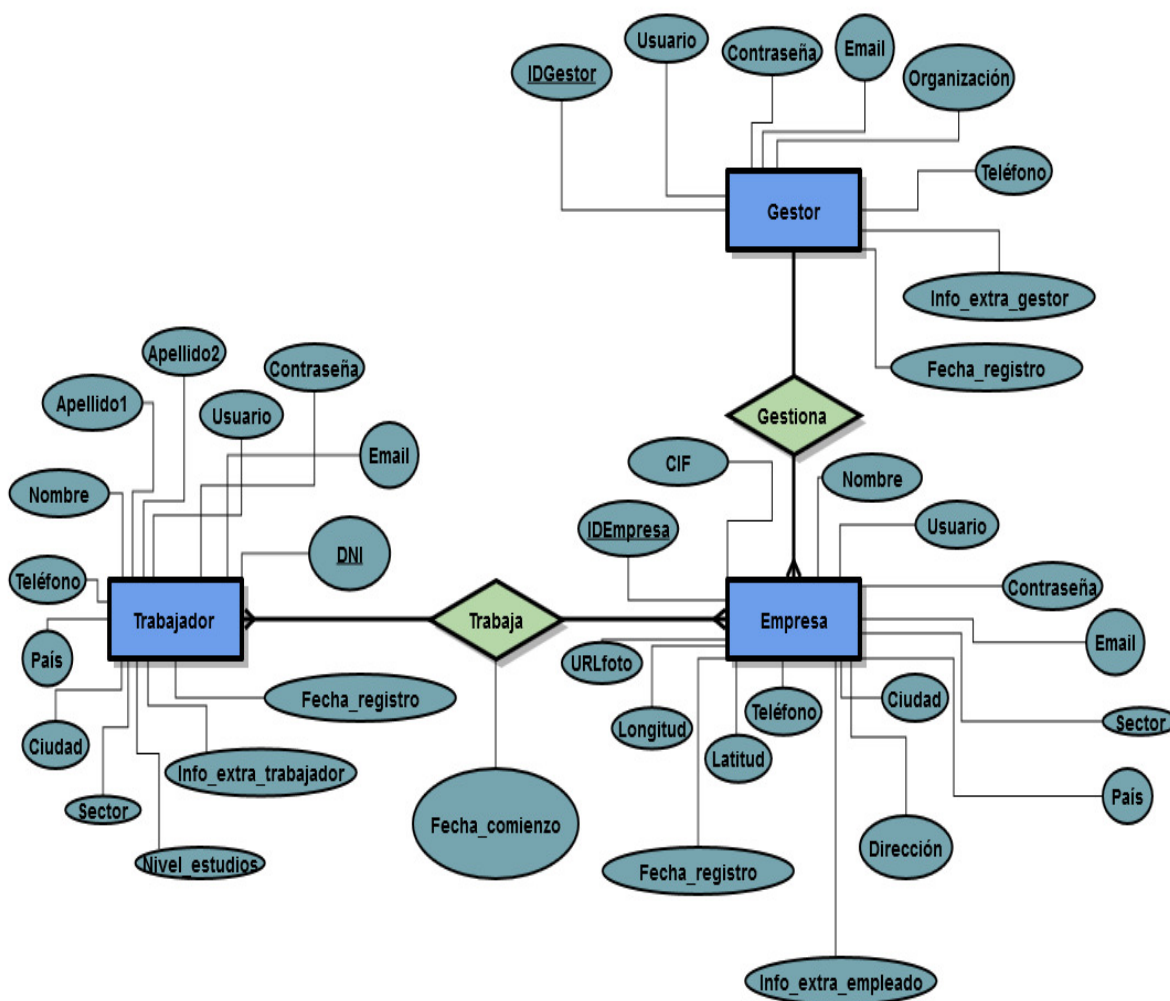
Detalla los medios en los que se almacena la información, como las particiones de disco. Esquema que se implementará en un manejador de bases de datos (DBMS).

5.1.1 Modelo entidad-relación

A continuación muestro el diseño entidad relación, que muestra el modo en que se ha configurado la estructura de la base de datos. Podemos observar que he creado tres entidades, correspondientes a cada uno de los posibles usuarios registrados de la aplicación web. Trabajador, Empresa y Gestor, que hace referencia a los gestores de empresas. Cada uno de ellos con sus atributos propios.

Entre trabajador y empresa se establece una relación de muchos a muchos, ya que un trabajador puede trabajar en muchas empresas a lo largo de su vida profesional y, a su vez, una empresa puede tener o haber tenido trabajando a diferentes empleados. Esto hace que genere una tabla intermedia entre Trabajador y Empresa de nombre “Trabaja” donde puedo almacenar los trabajadores que trabajan en qué empresas y he añadido un atributo explícito de la relación generada de nombre “Fecha_comienzo” que indica en qué fecha comenzó el trabajador a trabajar en dicha empresa.

Por otro lado, tenemos la relación entre Gestor Y Empresa que es una relación de cero a muchos, ya que un gestor puede crearse aún sin haber gestionado ninguna empresa, pero puede gestionar diferentes empresas. Por lo cual, siempre que un gestor cree una empresa, eso se registrará en la base de datos.

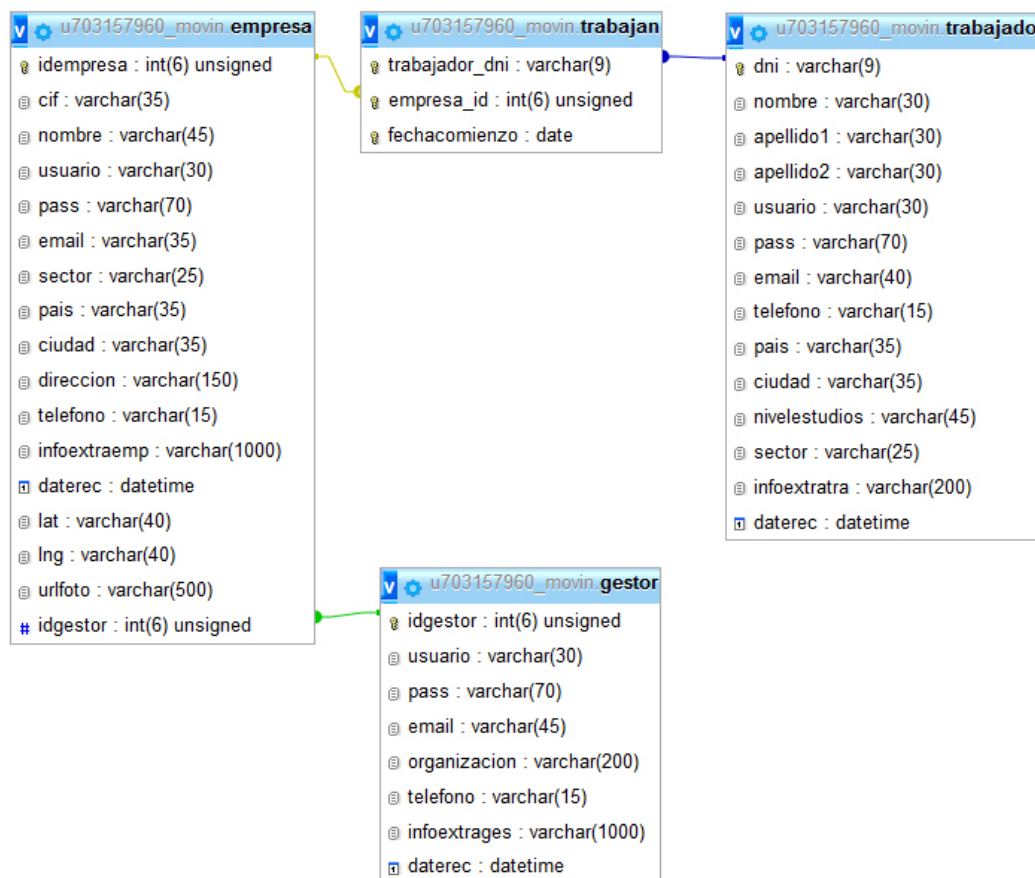


5.1.2 Modelo relacional

Observamos, a continuación, el modelo relacional de la base de datos. Esto indica cómo se han convertido las diferentes entidades y relaciones del modelo entidad relación anterior a las diferentes tablas y campos que tendremos en la base de datos MySQL. En la cual almacenaremos toda la información que nos interese. Además, podemos ver también el tipo de dato utilizado para almacenar los campos, los cual indica si utilizamos cadenas de caracteres, números, tipos de datos como fechas, etc.

Como campos de importancia reseñable podemos indicar “lat” y “lng” en la tabla “empresa”. Estos campos guardan las coordenadas de las empresas. Con estos datos se geolocalizan las empresas en el punto exacto del mapa. Además, es importante señalar que “empresa” tiene un atributo de nombre “idgestor” que es clave extranjera de “gestor”, lo cual indica que cada vez que un gestor registre una empresa, dicha empresa estará marcada por un código del gestor, facilitando esto consultas como encontrar todas las empresas un mismo gestor o incluso saber de qué gestor es cada empresa. También encontramos en la tabla “empresa” el campo “urlfoto” que se utiliza para guardar en dicho campo la cadena de caracteres que es el enlace a la imagen que la empresa quiere exhibir de sí misma.

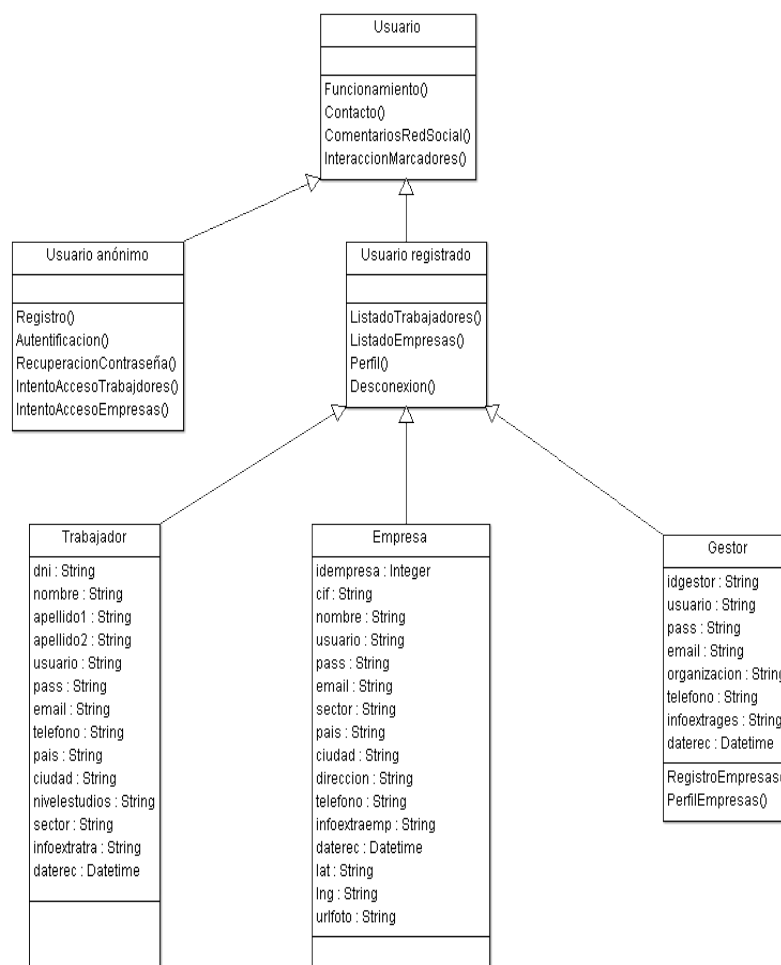
Vemos, además, cómo se ha formado la tabla “trabajan” entre “empresa” y “trabajador”, tema ya mencionado en el apartado anterior. Utilizando las claves primarias de “empresa” y “trabajador” así como el campo “fechacomienzo”.



5.2 Diagrama de clases

Muestro el diagrama de clases, donde podemos observar los diferentes usuarios que tendremos en la aplicación, las relaciones establecidas entre ellos (herencia) y los diferentes atributos y métodos a los que tienen acceso dichas clases.

Los métodos se encuentran explicados en el apartado 3.1.1 donde explicamos los requisitos funcionales. Ahí se describe qué deseamos conseguir con cada método. Además, los diferentes actores, en este caso las clases, son descritos en el apartado 3.2.1.

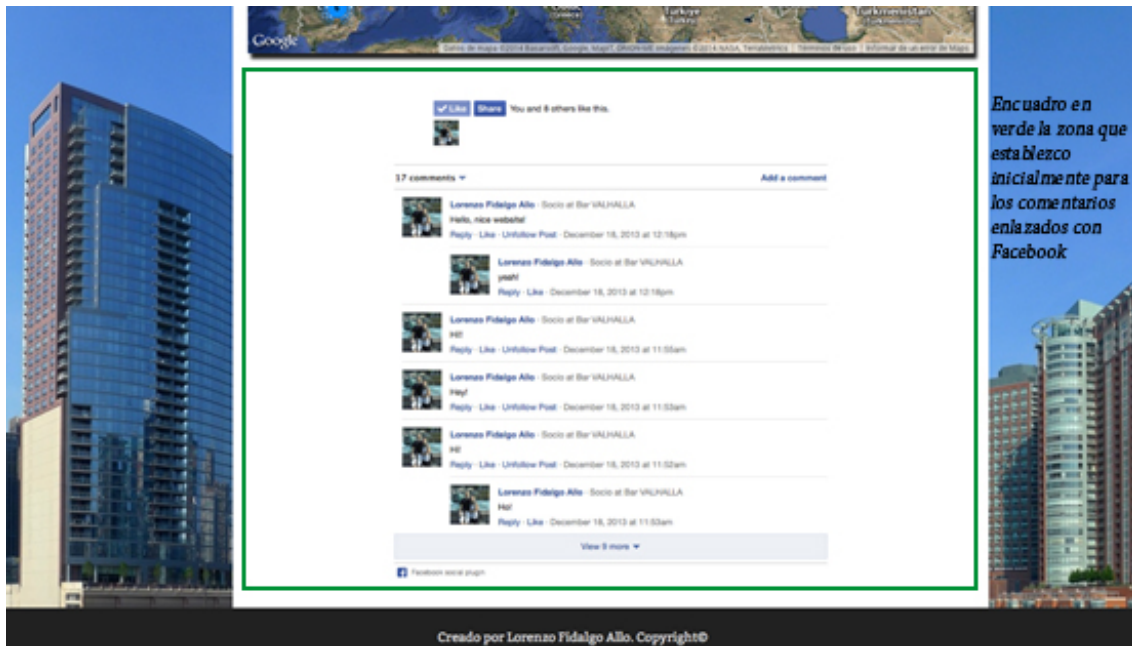


5.3 Diseño de interfaces

En los últimos años se observado una tendencia a aplicar una alta tasa de trabajo a una parte en concreto de los proyectos web. Ésta es el interfaz de usuario. Se trata del encargado de facilitar al usuario su labor a la hora de manipular la aplicación. El interfaz es realmente importante, ya que será el que consiga (o no) enganchar al usuario y lograr que forme parte de nuestra aplicación. Debe ser atractivo a primera vista y de fácil interacción, evitando causar problemas al cliente. Debe realizar su objetivo en el menor tiempo posible.

Procedo a mostrar la interfaz de usuario del proyecto web que nos ocupa, tratando de indicar las diferentes zonas observables en dicho proyecto. Para ello he tomado unas capturas de pantalla de una pequeña zona del proyecto web para mostrar la distribución de la página web.







Usuario

Contraseña

Tipo

Recuperar contraseña

¿Has olvidado tu contraseña de acceso?

Usuario

Tipo

En azul, el footer, estandarizado para toda la página.

Creado por Lorenzo Fidalgo Alfo. Copyright©

6. IMPLEMENTACIÓN

Sin duda, el módulo en el que he tenido que invertir más tiempo y ha hecho que el desarrollo se ralentice más ha sido trabajar con AJAX.

El haber incorporado la tecnología AJAX a lo largo de todo mi proyecto me ha otorgado muchos aspectos positivos de los que ya he hablado anteriormente en este mismo documento. Como contrapartida ha aumentado considerablemente las horas de desarrollo necesarias para la consecución de la aplicación. Me aventuraría a decir incluso que, como mínimo, ha cuádruplicado las horas necesarias para conseguir todos los requerimientos impuestos y autoimpuestos.

El porqué de aumentar de modo tan notable las horas necesarias de desarrollo va ligado con el cómo se implementa la citada tecnología en la aplicación, ya que si deseas, como era el caso que nos ocupa, relacionar la tecnología con absolutamente todo el funcionamiento, esto hace que la debas tener en cuenta en cada momento, modificando prácticamente todo el código que implementas. Muchas de las funciones implementadas en PHP o JavaScript cambian si estás trabajando con AJAX e incluso el cómo funciona el flujo de cajas generado con el lenguaje de marcado HTML cambia a causa de AJAX. Siendo una aplicación fácilmente exportable a otros lugares como aplicaciones de smartphones entre otros, me autoimpuse el requisito de que AJAX debía ser la espina dorsal del proyecto.

Ahora, como reflexión, a pesar de que el resultado final es totalmente satisfactorio, si no se tratase de un requisito obligatorio y en cambio se plantease incorporar en un proyecto de una empresa con varios empleados, se debería valorar si todo aquello que nos ofrece AJAX compensa con la cantidad de horas necesarias para la consecución de dicho requisito, ya que amplía considerablemente, como he mencionado constantemente, el número de horas de desarrollo necesarias para finalizar el proyecto.

Muestro, a continuación, imágenes en las que se puede observar la dimensión del código realizado para introducir AJAX en el proyecto. No estoy mostrando todo el código que ha debido ser creado o modificado para adaptarse a la tecnología AJAX, si no exclusivamente aquel código que es necesario para introducir AJAX allí donde ha interesado.

En el siguiente código tenemos parte de la codificación necesaria para convertir una petición normal de cierto área, como son en este caso la zona de contacto, de explicación del funcionamiento de la página, de la bolsa de trabajadores y de la bolsa de empresas en una carga de la zona central dinámica con peticiones AJAX.

```

submitteatiporegistro(); //Localizado en ajax

$('#contacto').click(function(event)
{
    event.preventDefault();
    fade();

    $.ajax(
    {
        url: 'contacto.html',
        success: function(data)
        {
            $('#rey').html(data);
            submittearmail();
        }
    });

});

$('#funcionamiento').click(function(event)
{
    event.preventDefault();
    fade();
    $.ajax(
    {
        url: 'funcionamiento.html',
        success: function(data)
        {
            $('#rey').html(data);
        }
    });
});

$('#trabajadores').click(function(event)
{
    event.preventDefault();
    fade();
    $.ajax(
    {
        url: 'trabajadores.php',
        success: function(data)
        {
            $('#rey').html(data);
        }
    });
});

$('#empresas').click(function(event)
{
    event.preventDefault();
    fade();
    $.ajax(
    {
        url: 'empresas.php',
        success: function(data)
        {
            $('#rey').html(data);
            submitteaideмп();
        }
    });
});

/* $('#home').click(function(event)
{
    event.preventDefault(); //LO HE COMENTADO
    $.ajax(

```

FiguraCódigo18 – Conversión a AJAX de algunas peticiones de contenido

Encontramos aquí más de las funciones necesarias para utilizar AJAX en nuestro proyecto. Además de utilizarlo para las zonas de perfil, recuperar contraseña y registro, aquí podemos encontrar los diferentes métodos necesarios para poder enviar a través de AJAX los formularios. Sin los cuales, sería imposible.

```
$('#perfil').click(function(event)
{
    event.preventDefault();
    fade();
    $.ajax(
    {
        url: 'perfil.php',
        success: function(data)
        {
            $('#rey').html(data);
            cambi();
            submiteagesup();
            submiteaempup();
            submiteatraup();
            submiteagesdempup();
            submiteatradelete();
            submiteaempdelete();
            submiteagesdelete();
            submiteagesdeletedemp();
            submiteacambi();
        }
    });
});

$('#recuperar_contrasena').click(function(event)
{
    event.preventDefault();
    fade();
    $.ajax(
    {
        url: 'recu_contra.html',
        success: function(data)
        {
            $('#rey').html(data);
            submitearecu();
        }
    });
});

$('#registro').click(function(event)
{
    event.preventDefault();
    fade();

    $.ajax(
    {
        url: 'registro.html',
        success: function(data)
        {
            $('#rey').html(data);
            submiteartra();
            submitearemp();
            submitearges();
        }
    });
});
```

FiguraCódigo19 – AJAX para formularios

El siguiente código realiza las funciones necesarias a la hora de que un usuario salga de su sesión de usuario. Genera los métodos AJAX necesarios para dar la posibilidad de que un usuario vuelva a identificarse.

```
function salir()
{
    $('#salir2').click(function(event)
    {
        event.preventDefault();

        $.ajax(
        {
            url: 'logout.php',
            success: function(data)
            {
                $('#log').html(data);

                $('#acceso').click(function(event)
                {
                    event.preventDefault();

                    $.ajax(
                    {
                        url: 'acceso.php',
                        success: function(data)
                        {
                            $('#rey').html(data);
                            submiteacom();
                        }
                    });
                });
            }
        });
    });
}

function saliendo()
{
    $('#salir').click(function(event)
    {
        event.preventDefault();

        $.ajax(
        {
            url: 'logout.php',
            success: function(data)
            {
                $('#log').html(data);

                $('#acceso').click(function(event)
                {
                    event.preventDefault();

                    $.ajax(
                    {
                        url: 'acceso.php',
                        success: function(data)
                        {
                            $('#rey').html(data);
                            submiteacom();
                        }
                    });
                });
            }
        });
    });
}
```

FiguraCódigo20 – AJAX para cambio de sesiones de usuarios

A continuación tenemos el código necesario para implementar AJAX en los formularios de envío de un email y, algo más complejo, los formularios correspondientes a la elección del tipo de usuario a registrar: Empresa, Trabajador, o Gestor.

```
function submitearmail()
{

    $(document).ready(function()
    {
        // Interceptamos el evento submit
        $('#formumail').submit(function()
        {
            // Enviamos el formulario usando AJAX
            $.ajax(
            {
                type: 'POST',
                url: $(this).attr('action'),
                data: $(this).serialize(),
                // Mostramos un mensaje con la respuesta
                success: function(data)
                {
                    $('#msj').html(data);
                }
            })
            return false;
        });
    });
};

function submiteatiporegistro()
{

    $(document).ready(function()
    {
        // Interceptamos el evento submit
        $('#formutiporegistro').submit(function()
        {
            // Enviamos el formulario usando AJAX
            //window.setTimeout("fade2",milliseconds);
            $.ajax(
            {
                type: 'POST',
                url: $(this).attr('action'),
                data: $(this).serialize(),

                // Mostramos un mensaje con la respuesta
                success: function(data)
                {
                    $('#rey').html(data);

                    submiteartra();
                    submitearemp();
                    submitearges();
                }
            })
            return false;
        });
    });
};
```

FiguraCódigo21 – AJAX para formulario de envío de email y registro de usuarios

Estas funciones AJAX sobre JQuery permiten el registro de los diferentes tipos de usuarios, ya que habilitan el envío de formularios a través de AJAX.

```
function submiteartra()
{
    $(document).ready(function()
    {
        // Interceptamos el evento submit
        $('#traform').submit(function()
        {
            // Enviamos el formulario usando AJAX
            $.ajax(
            {
                type: 'POST',
                url: $(this).attr('action'),
                data: $(this).serialize(),
                // Mostramos un mensaje con la r
                success: function(data)
                {
                    $('#msjrecord').html(data);
                }
            })
            return false;
        });
    });
};

function submitearemp()
{
    $(document).ready(function()
    {
        // Interceptamos el evento submit
        $('#empform').submit(function()
        {
            // Enviamos el formulario usando AJAX
            $.ajax(
            {
                type: 'POST',
                url: $(this).attr('action'),
                data: $(this).serialize(),
                // Mostramos un mensaje con la r
                success: function(data)
                {
                    $('#msjrecord').html(data);
                }
            })
            return false;
        });
    });
};

function submitearges()
{
    $(document).ready(function()
    {
        // Interceptamos el evento submit
        $('#gesform').submit(function()
        {
            // Enviamos el formulario usando AJAX
            $.ajax(
            {
                type: 'POST',
                url: $(this).attr('action'),
                data: $(this).serialize(),
                // Mostramos un mensaje con la r
                success: function(data)
                {
                    $('#msjrecord').html(data);
                }
            })
        });
    });
};
```

FiguraCódigo22 – Envío de formularios de registro a través de AJAX

Continuando con las imágenes anteriores, mostramos más código AJAX donde debemos habilitar distintos formularios y áreas para permitir el envío de datos con AJAX.

```
function submiteagesdeletedemp()
{
    $(document).ready(function()
    {
        // Interceptamos el evento submit
        $('#formgesdeleteodemp').submit(function()
        {
            // Enviamos el formulario usando AJAX
            $.ajax(
            {
                type: 'POST',
                url: $(this).attr('action'),
                data: $(this).serialize(),
                // Mostramos un mensaje con la respuesta de PHP
                success: function(data)
                {
                    $('#msjdeleteodemp').html(data);
                }
            })
            return false;
        });
    });
};

function submiteacom()
{
    $(document).ready(function()
    {
        // Interceptamos el evento submit
        $('#formacc').submit(function()
        {
            // Enviamos el formulario usando AJAX
            $.ajax(
            {
                type: 'POST',
                url: $(this).attr('action'),
                data: $(this).serialize(),
                // Mostramos un mensaje con la respuesta de PHP
                success: function(data)
                {
                    $('#log').html(data);
                    //Cambiamos la funcionalidad de "desconectarse"
                    $(document).ready(function()
                    {
                        $('#acceso').click(function(event)
                        {
                            event.preventDefault();

                            $.ajax(
                            {
                                url: 'acceso.php',
                                success: function(data)
                                {
                                    $('#rey').html(data);
                                    submiteacom();
                                }
                            });
                        });

                        $('#salir').click(function(event)
                        {
                            event.preventDefault();

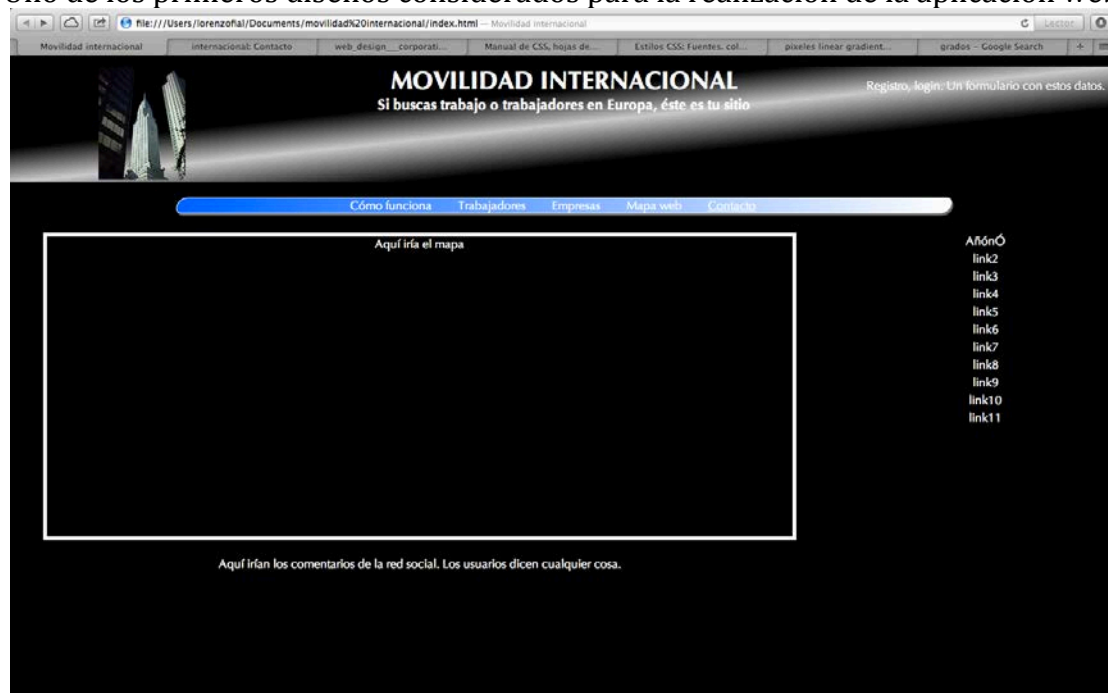
                            $.ajax(
                            {
                                url: 'logout.php',
                                success: function(data)
                                {
                                    $('#salir').html(data);
                                }
                            });
                        });
                    });
                }
            });
        });
    });
};
```

FiguraCódigo23 – Establecimiento de envío de datos de formularios con AJAX

Así mismo, otra de las áreas que me ha supuesto mayor dificultad, no por su complejidad, sino por la falta de aptitudes iniciales al comienzo del proyecto ha sido todo el tema de diseño gráfico que implica el proyecto, abarcando desde el layout o flujo de cajas, pasando por la coordinación de los diferentes colores en la aplicación web así como el diseño global de la página.

A través del estudio y la práctica conseguí mejorar en este aspecto y logré dignificar el proyecto, el cual, inicialmente, tenía a todas luces un muy mal diseño. Muestro, a continuación, algunos de los diseños que consideré inicialmente viables y que, desde luego, dejan mucho que desear. El diseño logrado finalmente no tiene nada que ver con éstos, siendo notablemente superior.

Uno de los primeros diseños considerados para la realización de la aplicación web.



Algo después del anterior, otro de los diseños planteados para la realización del proyecto.



Nada que ver con el diseño final del proyecto.



Por otro lado, una de las secciones que más he disfrutado a la vez que sufrido en su construcción es todo lo relacionado con el mapa del proyecto, la agrupación de los marcadores y la construcción de éstos, así como de las cajas de texto que le acompañan.

Prácticamente toda la complejidad de dicho módulo se encuentra codificada en lenguaje JavaScript. Inicialmente tuve algunos problemas con la creación dinámica de los marcadores, pero una vez solucioné esto y me las ingenié para crear y almacenar los datos de todas las empresas y sus marcadores en el mapa en la base de datos todo lo demás fue ir probando y disfrutar viendo el desarrollo del módulo. Por citar alguno, uno de los problemas que encontré en el desarrollo fue que google permite únicamente un número limitado de peticiones a google maps, plataforma en la que me apoyé para crear el mapa. Lo resolví almacenando en la base de datos unos parámetros de entrada que utilizo posteriormente en otras funciones para lograr de dicho modo crear de modo dinámico el mapa y todos los marcadores, sin necesitar, de este último modo, lanzar tantas peticiones a Google.

7. PRUEBAS Y RENDIMIENTO DEL SISTEMA

A través de todo el proyecto he realizado de modo continuado multitud de pruebas de rendimiento viendo cómo afectaban los diferentes cambios en el proyecto al resultado que observa el usuario. Debido a que nuestra aplicación no ha sido sobrecargada con archivos que necesiten grandes descargas de datos u otros inconvenientes del estilo, la navegación por el proyecto se produce de modo fluido y agradable para el usuario. Se han tenido en cuenta diferentes aspectos a la hora de tomar la decisión de implementar unas cosas u otras. Hablamos, por ejemplo, de las diferentes imágenes del proyecto. Todas ellas han sido tratadas con el programa Photoshop para conseguir reducir el peso de ellas y, como consecuencia, evitar crear retardo al usuario en el momento de ver la página. Se tuvo en consideración en todo momento la relación peso/calidad en las diferentes imágenes a la hora de insertar unas u otras en la aplicación web. Se ha evitado, a su vez, insertar efectos pesados que ralentizasen al usuario. Normalmente éstos podemos encontrarlos en JavaScript.

Inicialmente, se planteó la posibilidad de crear un efecto al clicar el usuario de la aplicación en un enlace para desplazarse a otra zona de la web. Se trataba de un efecto de desaparición de la zona central de la página web y creación de la sección en la cual el usuario estaba entrando, haciendo desaparecer lentamente el área en el cual el usuario se hallaba y haciendo aparecer al mismo tiempo la página en la cual el usuario va a entrar. Se observó, tras desarrollarlo, que este efecto hacía mucho más lenta la navegación del usuario, ya que debía esperar a que la página terminara de cargar, ya que el usuario estaba obligado a observar el bonito pero en ocasiones molesto efecto creado. Cuando el usuario ya era conocedor de las diferentes secciones de la aplicación y lo que quería era velocidad, dicho efecto la limitaba.

8. DESPLIEGUE

El despliegue del proyecto se realizó inicialmente en un servidor local de la máquina en la cual se estuvo desarrollando el proyecto en sus primeros pasos. Esto permitía probar el resultado de lo codificado, observar cómo se veía la aplicación en los diferentes navegadores y la fluidez de la aplicación en la navegación de los usuarios. Después de ser montado el proyecto en diferentes máquinas locales a través de XAMPP, se procedió a buscar un host y dominios gratuitos para poder desarrollar en internet la aplicación web con todo lo que ello conlleva.

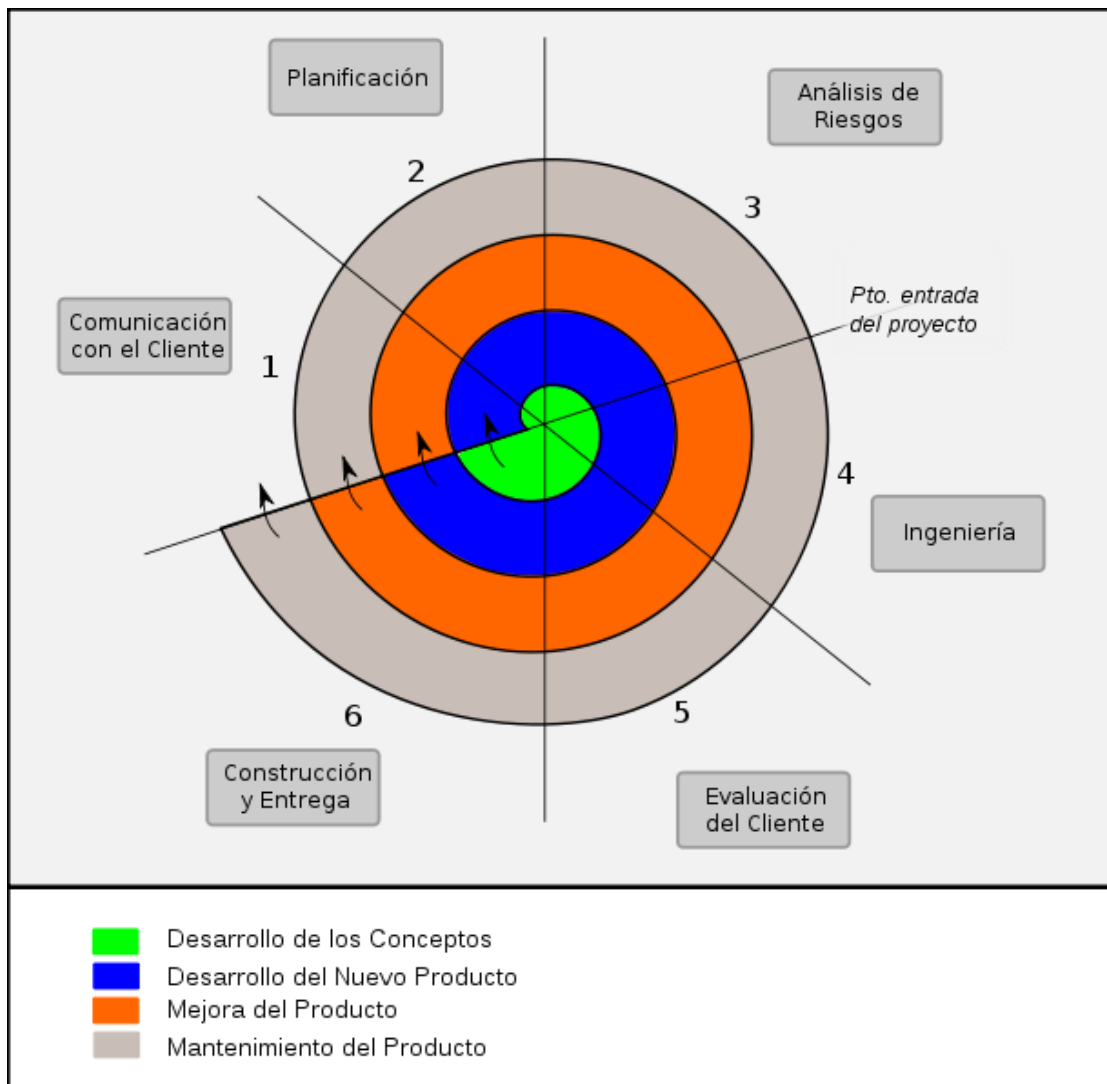
La base de datos fue manejada con phpmyadmin sobre MySQL.

La utilización de host y dominio gratuitos supusieron problemas a lo largo del desarrollo, ya que al ser gratuitos hay muchas opciones que no pueden utilizarse y debe uno adaptarse a esto resolviendo problemas que inicialmente no debería ser necesario afrontar. Concretamente, la utilización de un host gratuito implicó problemas a la hora de trabajar con MySQL, que finalmente pudieron ser resueltos. El host utilizado fue nixiweb y el dominio escogido fue movilidadint.tk. El uso de un dominio .tk supuso más problemas de los que aparentemente parecía que iba a causar. Los dominios .tk a lo largo de la historia de internet han sido utilizados en algunos casos para temas relacionados con compra-venta de drogas, apuestas ilegales y asuntos relacionados con éstos. Esto es así ya que no es necesario pagar por dichos dominios y no es necesario dar datos reales para poder trabajar con ellos.

Por todo lo citado anteriormente, se han establecido restricciones como la denegación de certificados gratuitos que son necesarios para la creación de enlaces a través de https, lo cual me causó problemas a la hora de enlazar la página web con Facebook. Además de esto, el dominio movilidadint.tk en el que tenía subido el proyecto dejó de funcionar ya que la empresa que manejaba los dominios .tk dejó de permitirme el acceso gratuito. Incluso en este momento es desaconsejable entrar en dicha dirección, advirtiendo los antivirus inseguridades en la página web. Finalmente, utilicé, además de la máquina local, otro servidor .tk para realizar las pruebas, de nuevo gratuito, pero ya prevenido de lo que podría ocurrir.

9. METODOLOGÍA DE GESTIÓN DEL PROYECTO

En el planteamiento del proyecto consideré muchas formas de afrontarlo. Una de ellas fue realizar la planificación completa del proyecto teniendo en cuenta los requisitos iniciales que mi tutor Jesús Villadangos y yo mismo instaurábamos en el proyecto y realizar directamente el diseño. Tras pensarlo detenidamente me pareció arriesgado y decidí seguir un planteamiento siguiendo el modelo en espiral de Boehm. Este modelo me proporcionó la estructura necesaria para poder realizar continuamente el proyecto sin sufrir demasiado por los cambios en los requisitos. Cambios producidos por elección propia o a través de sugerencias de módulos que mi tutor me sugirió implementar.



En el modelo en espiral, el software se desarrolla en una serie de versiones incrementales. Durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o un prototipo. Durante las últimas iteraciones, se producen versiones cada vez más completas del sistema diseñado. Pero cada nueva iteración aumenta más el coste del desarrollo.

Utilizar este modelo fue fundamental para el desarrollo del software, ya que en multitud de casos terminaba lo que parecía ser el objetivo final del proyecto y tras ello aparecían nuevos requisitos una y otra vez, ya que se deseaba mejorar la aplicación web y se iban añadiendo más módulos y funcionalidades cada vez o incluso perfeccionando éstas. Todo esto generaba una nueva iteración. Una nueva planificación de la ampliación, se fijaban los objetivos, análisis de riesgos posibles que generase esta nueva iteración, se desarrollaba y se observaba el resultado final de la iteración. En el caso concreto del proyecto que nos ocupa se realizaron multitud de iteraciones ya que los clientes (mi tutor y yo mismo) éramos exigentes y cada vez queríamos implementar nuevas funcionalidades, ya que el coste del proyecto y por ello de cada nueva iteración era mi propio trabajo, las horas que yo quisiera utilizar en la aplicación.

10. CONCLUSIONES

Tras muchísimas horas de planificación, aprendizaje, investigación y desarrollo me aventuro a asegurar que ha sido un proceso fructífero en el cual no sólo he conseguido crear el proyecto funcional en sí, ciñéndome a los requisitos planteados y ampliarlos sino que he logrado obtener los conocimientos y las aptitudes que me permiten desarrollar proyectos web completamente dinámicos y competitivos, aprendiendo a lo largo del camino lenguajes como HTML, CSS, PHP, o JavaScript. Lenguajes que, profesionalmente, son señaladamente importantes para el desarrollo de cualquier proyecto que tenga que ver con aplicaciones web, y no sólo con éstas, también con otras áreas como, por citar alguna, desarrollo de aplicaciones móviles, sector en un notable auge en estos momentos.

Analizando a posteriori el total desarrollo del proyecto podría mencionar que, pese a que la cantidad de horas dedicadas a la planificación del proyecto no fueron pocas en absoluto, quizá hubiera sido más eficiente haber invertido más horas incluso en dicha planificación, dado que en algunos momentos ha habido que revisar la aplicación web por falta inicial de conocimientos para realizar algún módulo y empezar con demasiada prontitud el desarrollo del software o incluso por cambio o inclusión de nuevos requisitos en el proyecto.

La consecución de los objetivos ha sido exitosa y el proyecto ha sido concluido satisfactoriamente, haciendo que la gran cantidad de horas utilizadas para la investigación y realización del proyecto hayan sido, sin duda, bien empleadas.

11. LÍNEAS FUTURAS

Dado el punto actual del proyecto, los siguientes pasos deberían ser dirigidos a dar a conocer la aplicación y ganar su propio nicho de mercado. Se debería hacer saber a las empresas de la existencia del proyecto y una vez habiendo conseguido tener a diferentes empresas de Europa en nuestra base de datos, utilizar a éstas para comenzar a trabajar en la bolsa de trabajadores que buscan empleo. El solo hecho de tener a las citadas empresas en la bolsa de nuestra aplicación conseguirá atraer por sí solo a diferentes trabajadores y éstos a su vez a otros. Tras esto, o incluso paralelamente debería maximizarse el uso de las redes sociales como Facebook principalmente aprovechando la conexión directa desarrollada en la propia aplicación y otras redes sociales como twitter, considerando la posibilidades de incluir un nexo entre twitter y nuestra propia web. Una vez observada la retroalimentación que se conseguiría observando todos los comentarios de los usuarios nos plantearíamos la creación, modificación o eliminación de diferentes módulos del proyecto. La obtención de dicho feedback es realmente valiosa, ya que esta información es, precisamente, aquello que nos indica a dónde debemos dirigidos para otorgar a los usuarios de nuestro proyecto lo que ellos desean, lo cual debería ser, en todo momento, nuestro objetivo.

Además de todo lo citado anteriormente, también debería ser contratado un Host y dominio de pago que generen seguridad para no tener problemas en un futuro cercano. Seguramente se buscaría un dominio .com.

Por otro lado y por último, mencionar que otro paso importante debería ser crear una aplicación para móviles utilizando herramientas multiplataforma desde la propia página web o incluso crear las aplicaciones nativas en Android e IOS.

12. BIBLIOGRAFÍA

Una de las características que me veo en la obligación de citar en esta memoria es la amplísima bibliografía utilizada en la creación de este proyecto.

Esto es debido a la gran cantidad de áreas que han sido utilizadas para el desarrollo de la aplicación. No sólo han sido utilizados multitud de libros de referencia e incluso de aprendizaje desde cero de diferentes tecnologías.

Algunos de ellos han sido:

1. HTML5 y CSS3: Revolucione el diseño de sus sitios web
2. El gran libro de HTML5, CSS3 y JavaScript
3. Mi primer proyecto Web
4. Domine JavaScript
5. PHP práctico. Fundamentos.



Sino que también han sido utilizados muchísimos recursos de internet. Por citar algunos de gran importancia podría ser la plataforma de Youtube “Mejorandola” que habla acerca de muchísimas tecnologías informáticas o la página web “desarrolloweb.com” que ha sido muy importante y de gran referencia a lo largo de la construcción del proyecto.

